



UNIVERSIDAD DE VALLADOLID

FACULTAD DE MEDICINA

ESCUELA DE INGENIERÍAS INDUSTRIALES

TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA BIOMÉDICA

**Adquisición de base de datos de movimientos
de miembro superior en adulto sanos y
pacientes pediátricos con parálisis cerebral.
Clasificación de movimientos con visión
artificial y aprendizaje profundo.**

Autora:

D.^a Carla de Gregorio Bermejo

Tutor:

D. Mario Martínez Zarzuela

D.^a Cristina Simón Martínez

Valladolid, 10 de julio de 2023

TÍTULO: Adquisición de base de datos de movimiento de miembro superior en adulto sanos y pacientes pediátricos con parálisis cerebral. Clasificación de movimientos con visión artificial y aprendizaje profundo.

AUTOR/A: **D.ª Carla de Gregorio Bermejo**

TUTOR/A: **D. Mario Martínez Zarzuela**
D.ª Cristina Simón Martínez

DEPARTAMENTO: **Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática (TSCIT)**

TRIBUNAL

PRESIDENTE: **D. Carlos Gómez Peña**

SECRETARIO: **D. Mario Martínez Zarzuela**

VOCAL: **D. Daniel Álvarez González**

SUPLENTE 1: **D. Alberto Mansilla Gallo**

SUPLENTE 2: **D. Jesús Poza Crespo**

FECHA: 10 de julio de 2023

CALIFICACIÓN:

ÍNDICE

RESUMEN.....	vi
ABSTRACT	vii
AGRADECIMIENTOS	viii
LISTA DE FIGURAS	ix
LISTA DE TABLAS.....	xi
INTRODUCCIÓN.....	1
Contexto y motivación.....	1
Hipótesis y objetivos	2
Estudios previos.....	3
I. Sensores inerciales.....	3
II. Visión artificial.....	4
III. Fusión sensores inerciales con visión artificial	5
1. PARÁLISIS CEREBRAL.....	6
1.1 Introducción.....	6
1.2 Etiología.....	6
1.3 Clasificación	7
1.4 Escala para la evaluación del miembro superior.....	8
1.4.1 The Manual Ability Classification System (MACS).....	8
1.4.2 The Assisting Hand Assessment (AHA)	9
1.4.3 The Melbourne Assessment (MA).....	9
2. SENSORES INERCIALES: XSENS AWINDA	11
2.1 Introducción a los sensores inerciales.....	11
2.2 Introducción XSens Awinda.....	11
2.3 Colocación sensores.....	12
2.4 Calibración sensores.....	14
2.5 Grabación	15
3. VISIÓN ARTIFICIAL: OPENCAP	16
3.1 Introducción a la visión artificial	16
3.2 Introducción OpenCap	16
3.3 Conceptos teóricos acerca del funcionamiento.....	16
3.4 Calibración.....	17
3.5 Grabación	19

3.6	Comparación con otros métodos.....	20
4.	DEEP LEARNING.....	22
4.1	Introducción al Deep Learning.....	22
4.2	Redes neuronales.....	23
4.3	Redes convolucionales.....	23
4.4	Autoenconders.....	24
4.5	Técnicas de regularización.....	25
4.6	Métricas de rendimiento en redes neuronales.....	26
5.	MATERIALES Y MÉTODOS.....	28
5.1	Bases de datos.....	28
5.2	Materiales.....	28
5.3	Participantes.....	29
5.4	Comité de Ética.....	31
5.5	Protocolo.....	32
5.6	Bloque I: Reconocimiento de actividades de la vida diaria.....	34
5.6.1	Introducción datos OpenCap.....	34
5.6.2	Movimientos de interés.....	35
5.6.3	Preprocesado de datos.....	37
5.7	Bloque II: Segmentación de las fases de “Open/Close a bottle and pour water”.....	37
5.7.1	Introducción y fases de Open/Close a bottle and pour water.....	37
5.7.2	Etiquetado de datos.....	40
5.7.3	Preprocesado de datos.....	42
6.	RESULTADOS Y DISCUSIÓN.....	43
6.1	Resultados.....	43
6.1.1	Bloque I: Reconocimiento de actividades de la vida diaria.....	43
6.1.2	Bloque II: Segmentación de las fases del movimiento Open/Close a bottle and pour water.....	52
6.2	Discusión.....	82
6.2.1	Bloque I: Reconocimiento de actividades de la vida diaria.....	82
6.2.2	Bloque II: Segmentación de las fases del movimiento Open/Close a bottle and pour water.....	83
6.3	Limitaciones.....	85
6.4	Problemas.....	86
7.	CONCLUSIONES Y LÍNEAS FUTURAS.....	87

7.1	Conclusiones.....	87
7.2	Líneas futuras.....	87
	BIBLIOGRAFÍA	90

RESUMEN

La parálisis cerebral es la causa más frecuente de discapacidad motora en los niños, afectando a 2-3/1000 recién nacidos. Esta patología es un trastorno no progresivo causado por lesiones cerebrales en el periodo prenatal, perinatal y postnatal.

Los trastornos motores suelen ir acompañados de alteraciones de la sensibilidad, la percepción, la cognición, la comunicación y el comportamiento, como la epilepsia y problemas musculoesqueléticos secundarios.

La fisioterapia promueve el mantenimiento y la rehabilitación de los pacientes para su bienestar psicológico y social. Muchas terapias están incluyendo avances tecnológicos, cambiando el concepto de salud y ayudando a estos pacientes.

El objetivo principal del proyecto es la adquisición de una base de datos de actividades de miembro superior y el desarrollo de un sistema de aprendizaje automático para el reconocimiento de acciones humanas de miembro superior. Este conjunto de tareas son unimanuales, bimanuales o combinación de ambas, siendo actividades muy presentes en la vida diaria del paciente.

En primer lugar, se define un protocolo de adquisición de movimientos y se registrarán los movimientos de los sujetos empleando sensores inerciales comerciales XSens Awinda y el *software* de visión artificial de acceso abierto OpenCap. A continuación, se analizan y preprocesan los datos capturados, para un posterior entrenamiento con redes neuronales. Este proceso se realiza con la finalidad de reconocer las diversas actividades de la vida diaria capturadas en sujetos sanos y en pacientes.

Adicionalmente al reconocimiento de actividades, se estudia la segmentación de las fases de un movimiento más complejo. Para ello es necesario un etiquetado manual de las fases del movimiento y nuevas pruebas de entrenamiento con redes neuronales. En esta fase del proyecto se emplean nuevamente los datos de sujetos sanos y en pacientes.

ABSTRACT

Cerebral palsy is the most common cause of motor disability in children, affecting 2-3/1000 newborns. Cerebral palsy is a non-progressive disorder caused by prenatal, perinatal, and postnatal brain damage.

Motor disorders are often accompanied by alterations in sensation, perception, cognition, communication, and behaviour, such as epilepsy and secondary musculoskeletal problems.

Physical therapy promotes the maintenance and rehabilitation of the patients for their psychological and social well – being. Many therapies are including technological advances, changing the concept of health, and helping these patients.

The main objective of the project is the acquisition of an upper limb activity database using commercial inertial sensors XSens Awinda and the open access artificial vision software OpenCap.

Among the sub-objectives of this final degree project is the development of a machine learning system for the recognition of human upper limb actions. This set of tasks are unimanual, bimanual or a combination of both, being activities very present in the patient's daily life.

The first part of the project consists of recording control subjects and paediatric patients performing these movements. The videos will then be analysed and pre-processed for further training with neural networks. This process will be carried out in order to identify the various activities captured in healthy subjects and patients.

In addition, another sub-objective is the segmentation of the phases of a complex movement. Starting with a manual labelling of the captured phases, followed by a pre-processing. It will end with the training of a deeper neural network than the one previously used. In this phase of the project, healthy, paediatric, and pathological subjects will again be used.

AGRADECIMIENTOS

A todos los que han formado parte de esta etapa.

A mi familia, en especial a mis padres, por acompañarme
y apoyarme en todo lo posible y mucho más.

Y sobre todo a Mario, por todo tu apoyo y por hacer posible este TFG.

Gracias por transmitirme tu pasión y tus conocimientos sobre este tema.

“Solo con el corazón se puede ver bien,
lo esencial es invisible a los ojos”
Antoine de Saint-Exupéry

LISTA DE FIGURAS

Figura 1. Factores de riesgo potenciales de la parálisis cerebral (Sadowska et al., 2020)	7
Figura 2. Maletín del Sistema XSens Awinda junto con sus sensores inerciales (Movella, 2022).....	12
Figura 3. Sensores inerciales del Sistema XSens Awinda.....	12
Figura 4. Sensor de la cabeza del Sistema XSens Awinda (Movella, 2022)	13
Figura 5. Sensores de los hombros del Sistema XSens Awinda	13
Figura 6. Sensores de las extremidades superiores del Sistema XSens Awinda	14
Figura 7. Posición neutral de calibración del Sistema XSens Awinda	15
Figura 8. Tablero de calibración de OpenCap	18
Figura 9. Disposición del tablero y las cámaras.....	18
Figura 10. Posición neutral de calibración de OpenCap	19
Figura 11. Código QR para sincronizar las cámaras de OpenCap	20
Figura 12. Captura de movimiento basada en marcadores (MOCAP) comparado con análisis basado en vídeo (OpenCap) (Uhlrich et al., 2022b).....	20
Figura 13. Técnicas del análisis de datos (Choi et al., 2020).....	23
Figura 14. Estructura de un autoencoder (Pathirage et al., 2018)	24
Figura 15. Libro empleado para el movimiento "Turn pages" y sombrero empleado para el movimiento de "Put on a hat/helmet"	29
Figura 16. Botellas y vaso empleados para el movimiento "Open/Close a bottle and pour water "	29
Figura 17. Sala "Leonardo Torres Quevedo"	32
Figura 18. Colocación sensores del Sistema XSens Awinda para miembro superior	33
Figura 19. Tablero de calibración OpenCap con vectores X, Y y Z.....	34
Figura 20. Interfaz desarrollada para el etiquetado de datos	40
Figura 21. Frame de la interfaz con los botones de reproducción de los vídeos.	40
Figura 22. Frame de la interfaz con los botones de etiquetado.....	41
Figura 23. Frame de la interfaz para etiquetar a los sujetos patológicos	41
Figura 24. Red neuronal convolucional de 4 capas de 32, 64, 128 y 256 filtros.....	44
Figura 25. Matriz de confusión de la Tabla 5	45
Figura 26. Matriz de confusión de la Tabla 6.	47
Figura 27. Matriz de confusión de la Tabla 7.	48
Figura 28. Matriz de confusión de la Tabla 8.	49
Figura 29. Red neuronal con 4 capas Dense de 32, 64, 128 y 32 neuronas.	53
Figura 30. Red neuronal 4 capas Dense de 32, 64, 128 y 256 neuronas.	54
Figura 31. Matriz de confusión de la Tabla 10.....	55
Figura 32. Red neuronal de 4 capas Dense de 32, 64, 128 y 32 neuronas.....	56
Figura 33. Red neuronal de 4 capas Dense de 32, 64, 128 y 256 neuronas.....	57
Figura 34. Matriz de confusión de la Tabla 11.....	58
Figura 35. Red neuronal con 6 capas Dense y con DropOut	59
Figura 36. Matriz de confusión de la Tabla 12.....	60
Figura 37. Red neuronal con 7 capas Dense y con DropOut, imitando arquitectura de "cono invertido"	62

Figura 38. Red neuronal convolucional con 4 capas y con DropOut.....	63
Figura 39. Red neuronal convolucional empleando DropOut y BatchNormalization	64
Figura 40. Red neuronal convolucional usando DropOut, BatchNormalization y regularización L1 y L2.....	65
Figura 41. Red neuronal convolucional imitando la arquitectura de “cono invertido”	66
Figura 42. Red neuronal convolucional empleando Global Average Pooling.....	67
Figura 43. Matriz de confusión de la Tabla 14.....	69
Figura 44. Red neuronal convolucional con 2 capas Dense finales de 128 y 64 neuronas.....	71
Figura 45. Matriz de confusión de la Tabla 17.....	74
Figura 46. Matriz de confusión de la Tabla 18.....	75
Figura 47. Matriz de confusión de la Tabla 19.....	77
Figura 48. Matriz de confusión de la Tabla 20.....	78

LISTA DE TABLAS

Tabla 1. Sujetos sanos participantes.	30
Tabla 2. Sujetos patológicos participantes.....	31
Tabla 3. Movimientos seleccionados de miembro superior.....	36
Tabla 4. Fases y límites propuestos para la segmentación del movimiento Open/Close a bottle and pour water	39
Tabla 5. Métricas de rendimiento del entrenamiento de sujetos controles para todos los movimientos	45
Tabla 6. Métricas de rendimientos pediátricos con todos los movimientos.	46
Tabla 7. Métricas de rendimiento del K-Fold realizado a los sujetos pediátricos.	47
Tabla 8. Métricas de rendimiento de entrenamiento sujetos controles y patológicos.	49
Tabla 9. Resumen de los resultados de los experimentos Bloque I: Reconocimiento de actividades de la vida diaria.	51
Tabla 10. Métricas de rendimiento con ventanas de 30 frames.....	54
Tabla 11. Métricas de rendimiento con ventana de 15 frames.....	57
Tabla 12. Métricas de rendimiento de ventana de 45 frames.	59
Tabla 13. Análisis de patrones de movimiento de sujetos adultos sanos.....	61
Tabla 14. Métricas de rendimiento de red neurona convolucional con capa de Global Average Pooling.....	68
Tabla 15. Análisis de patrones de movimiento de sujetos pediátricos.....	72
Tabla 16. Patologías de los sujetos pediátricos.....	73
Tabla 17. Métricas de rendimiento de entrenamiento con sujetos patológicos.....	74
Tabla 18. Métricas de rendimiento de entrenamiento con sujetos pediátricos sanos y patológicos.....	75
Tabla 19. Métricas de rendimiento de entrenamiento con sujetos sanos y patológicos.....	76
Tabla 20. Métricas de rendimiento con la etiqueta verter.....	77
Tabla 21. Resumen I de los resultados de los experimentos del Bloque II: segmentación de las fases del movimiento Open/Close a bottle and pour wáter....	79
Tabla 22. Resumen II de los resultados de los experimentos del Bloque II: segmentación de las fases del movimiento Open/Close a bottle and pour wáter....	80
Tabla 23. Resumen III de los resultados de los experimentos del Bloque II: segmentación de las fases del movimiento Open/Close a bottle and pour wáter....	81

INTRODUCCIÓN

Contexto y motivación

El análisis del movimiento humano tiene gran importancia en el contexto de la Ingeniería Biomédica, siendo un campo desafiante dentro de la investigación científica. En los últimos años, ha despertado un interés creciente en áreas como la biomecánica, la rehabilitación, la fisioterapia deportiva, la robótica, proporcionando grandes avances, tanto en su análisis como en el desarrollo de sistemas de captura del movimiento (Armitano-Lago et al., 2022)

Tiene especial relevancia comprender la cinemática del movimiento humano para evaluar el rendimiento funcional de las extremidades, tanto en condiciones normales como patológicas. Además, para el diagnóstico adecuado y la valoración de la efectividad de ciertos tratamientos de enfermedades articulares y musculares, así como, el diseño de dispositivos protésicos (Moya Jiménez, 2023).

La parálisis cerebral es la causa más frecuente de discapacidad motora en los niños, afectando a 2-3/1000 nacidos, hasta 40-100/1000 entre los bebés prematuros. Esta patología es un trastorno no progresivo causado por lesiones cerebrales en el periodo prenatal, perinatal y postnatal.

Los trastornos motores suelen ir acompañados de alteraciones de la sensibilidad, la percepción, la cognición, la comunicación y el comportamiento, como la epilepsia y problemas musculoesqueléticos secundarios (Elena Beani et al, 2019).

En el tratamiento de la parálisis cerebral, a menudo los médicos y especialistas analizan de forma cualitativa y a partir de su experiencia previa la evolución de los niños. Debido a esto, se busca una alternativa para valorar de forma cuantitativa y objetiva a los pacientes, sin que sea dependiente de los conocimientos del profesional.

Los sistemas de captura de movimiento se han convertido en una herramienta prometedora para evaluar el movimiento, tanto en entornos naturales como clínicos. Facilitan la comprensión, así como, el desarrollo de bases de datos del movimiento humano (Mathis & Mathis, 2020).

La recopilación de datos se puede llevar a cabo a través de sensores vestibles, como unidades de medición inercial (IMU) conectadas a las extremidades del cuerpo para estimar poses 3D precisas (von Marcard et al., 2018). Obteniendo información de alta calidad de la forma, la postura y la cinemática del cuerpo humano, pudiendo utilizarse para el reconocimiento de acciones, el modelado de movimientos, el análisis de la marcha y la reconstrucción de la forma del cuerpo (Ghorbani et al., 2021).

También, se pueden emplear cámaras, desde dispositivos muy sofisticados a cámaras de teléfonos portátiles. Variando el número de dispositivos que se emplean

en las grabaciones, desde una cámara con un punto de vista, hasta grabaciones con 2-3 cámaras sincronizadas, pudiendo realizar reconstrucciones 3D del sujeto de interés (Ghorbani et al., 2021).

Las investigaciones que emplean sensores inerciales vestibles con cámaras de dispositivos portátiles se enfocan en el análisis y estudio de la marcha humana. Por lo que, las bases de datos que recogen información del miembro superior son escasas, pudiendo ser interesante analizar dichas bases. Muchos pacientes con parálisis cerebral presentan afectación del miembro superior, interfiriendo en su autonomía en la realización de tareas de la vida diaria en su entorno natural.

Este estudio se enmarca en un proyecto de investigación a largo plazo, cuyo objetivo es conseguir una terapia de rehabilitación individualizada, tanto del miembro superior como inferior. Empleando dichos dispositivos, facilitando el proceso de rehabilitación en un entorno conocido y cómodo para el paciente, como puede ser su domicilio.

Hipótesis y objetivos

La hipótesis de este Trabajo Fin de Grado es, si es posible reconocer movimientos relacionados con actividades de la vida diaria en pacientes pediátricos con parálisis cerebral, empleando dispositivos de adquisición que en el futuro podrán ser utilizados en el domicilio de los pacientes.

Uno de los principales objetivos del proyecto final, es la adquisición de una base de datos de actividades de miembro superior empleando sensores inerciales, XSens Awinda y las cámaras de dos iPads como visión artificial. Se utilizará, además, el *software* de bajo coste y acceso abierto, OpenCap que, con los datos obtenidos de las cámaras, reconstruirá el esqueleto de los sujetos.

Dentro de los subobjetivos de este proyecto, es el desarrollo de un sistema de aprendizaje automático para el reconocimiento de actividades de miembro superior en sujetos sanos y pediátricos patológicos. Dichos movimientos son realizados en la vida diaria de los pacientes, siendo tareas unimanuales, bimanuales o combinación de ellas.

Otro de los objetivos propuestos es la segmentación de tareas dentro de una actividad compleja compuesta del miembro superior. Comenzando por un etiquetado manual de las fases capturadas, seguido de un preprocesamiento. Finalizando con un entrenamiento con redes neuronales para su posterior clasificación. Para conseguir dicho objetivo, se tratará de encontrar la arquitectura óptima para el reconocimiento de las actividades propuestas.

Se buscará alcanzar ambos objetivos, empleando sensores vestibles y sistemas de visión artificial, basados en dispositivos móviles, iPads y aplicaciones asequibles para la adquisición del movimiento o la recolección de datos. A su vez, la información

proporcionada por los dispositivos se utilizará en combinación con sistemas de aprendizaje automático y aprendizaje profundo para el reconocimiento y clasificación de actividades.

La población diana de este estudio, a la cual se capturará el movimiento del miembro superior usando los dispositivos anteriormente mencionados, serán participantes sanos tanto adultos como pediátricos; y pacientes pediátricos con parálisis cerebral.

Estudios previos

I. Sensores inerciales

Los sensores inerciales o Unidades de Medición Inercial (IMU) juegan un papel fundamental en la adquisición de datos de acción humana, pudiéndose utilizar posteriormente, como entrada para modelos neuronales.

En las investigaciones previas se incluyen los estudios enfocados en usar dichos dispositivos para el reconocimiento de acciones, desde las más simples como es el caso de (Elkholy et al., 2020) o más complejas, como deportes, como ocurre con (McGrath et al., 2021).

En relación con el primero, (Elkholy et al., 2020) emplean los sensores inerciales para clasificar seis actividades de las extremidades superiores humanas. Estas actividades consistían en: movimiento de flexión y extensión de brazos, pronación y supinación de brazos, rotaciones internas y externas de hombros.

Se llevó a cabo para investigar si estas actividades se realizaban de manera normal utilizando dos redes neuronales diferentes: una red neuronal artificial simple basada en capas *Dense* y una red neuronal convolucional.

En relación con el segundo, (McGrath et al., 2021) propusieron el uso de dichos dispositivos para el reconocimiento actividades específicas de deporte, centradas en el uso del miembro superior. Como el artículo anterior, de (Elkholy et al., 2020), utilizaron redes convolucionales para el reconocimiento de las actividades.

Finalmente, también se emplean para el reconocimiento de acciones relacionadas con el ámbito de la salud. A modo de ejemplo, (Mahmoud et al., 2021) propusieron un método de evaluación para el miembro superior a partir de sensores inerciales. Dicho procedimiento utilizaba información cinemática de los sensores para clasificar las acciones de rehabilitación después de un accidente cerebrovascular. También, se analizaba la calidad de las acciones durante las sesiones de rehabilitación, implementando una mayor efectividad en el tratamiento propuesto por los terapeutas.

En este caso, nuevamente, emplearon redes convolucionales para llevar a cabo dicha clasificación y evaluación de actividades.

II. Visión artificial

El reconocimiento del movimiento humano ha llamado la atención de investigadores del dominio de la visión artificial debido a su amplia variedad de aplicaciones, dentro de la robótica, la atención médica y la interacción humano-computadora.

Siendo una tarea desafiante, ya que, a diferencia de la clasificación de imágenes fijas, los datos de vídeo contienen información temporal, que juega un papel fundamental a la hora reconocer las acciones (Wu et al., 2017).

En los últimos años, se ha trabajado en diferentes áreas dentro de la visión artificial, como la clasificación de vídeo, la identificación de acciones y la segmentación. Generalmente, un sistema de reconocimiento de acciones analiza ciertas secuencias de vídeo o ventanas para aprender los patrones de interés de las acciones humanas (Wu et al., 2017)

Dentro de los estudios previos, se encuentran los artículos que se focalizan en la adquisición y reconocimiento del miembro superior, como (Cao et al., 2019; Kumar & Kumar, 2023). Por último, dentro de la visión artificial aplicada a la medicina, en especial a pacientes pediátricos, se encuentra (Leo et al., 2022), que en particular, capturan y analizan el movimiento del cuerpo entero, tanto extremidad superior como inferior.

En relación con los primeros, (Cao et al., 2019) realizaron una adquisición de secuencias de acción, diseñadas por un terapeuta profesional. Estas secuencias se focalizaban en el miembro superior de pacientes que había sufrido una lesión en la médula espinal. Dicha captura de movimiento se realizó con el sensor Kinect, el cual lleva integrada una cámara, que recoge los datos de movimiento.

La finalidad de este estudio fue la evaluación de dichos movimientos, según su grado de afectación en las extremidades superiores, mediante una red neuronal combinada, una red Long Short-Term Memory y una red neuronal convolucional.

La captura de actividades de las extremidades superiores, también se llevó a cabo en (Kumar & Kumar, 2023). En este caso, se adquirió una base de datos, de movimientos de miembro superior, extrayendo un total de 50 clases de acción en vídeos cortos. Propusieron una arquitectura de red basada en enconder, para agrupar actividades similares, clasificándolas por grupos.

En relación con el segundo grupo, dentro del reconocimiento de acciones humanas en pacientes pediátricos, (Leo et al., 2022), buscaban el diagnóstico temprano de trastornos de neurodesarrollo en bebés. En este caso, las acciones propuestas eran: gatear, sentarse, pararse y caminar, grabadas a través de cámaras Kinect.

Posteriormente, trataron de desarrollar un método automático, entrenado con dichas acciones, para evaluar las habilidades de movilidad de los niños, que recientemente habían aprendido a caminar. La red neuronal estaba formada por la combinación de redes convolucionales y redes Long Short-Term Memory.

III. Fusión sensores inerciales con visión artificial

Entre las investigaciones previas, que relacionan adquisiciones de datos empleando sensores inerciales y cámaras de visión artificial, destaca el estudio de (Ghorbani et al., 2021).

En dicho estudio, capturan el movimiento del cuerpo humano a través de un sistema de captura de movimiento óptico y sensores de unidad de medición inercial. También, emplean dos cámaras de vídeo portátiles, es decir, dos smartphones.

Los participantes de dicho estudio pasaron por 5 secuencias de captura de datos, realizando un total de 20 movimientos. Las actividades realizadas abarcaban desde movimientos simples básicos, como pronosupinación y flexión-extensión del brazo, hasta movimientos más complejos, como ponerse un sombrero.

El conjunto de datos que recopilaron (Ghorbani et al., 2021), es de gran interés al combinar información de sensores inerciales, captura de movimiento y vídeo de manera sincronizada y calibrada, brindando una representación precisa de la forma y la postura corporal en 3D.

1. PARÁLISIS CEREBRAL

1.1 Introducción

La parálisis cerebral es la causa más común de discapacidad motora infantil, siendo un problema de salud pública a nivel mundial, afectando a 2-3/1000 recién nacidos, hasta 40-100/1000 entre los bebés prematuros (Elena Beani et al, 2019).

Esta descrita como un conjunto de trastornos que afectan al desarrollo del movimiento y la postura, durante el desempeño de las actividades básicas diarias. Resulta de una lesión en el cerebro en desarrollo, ocurre a una edad temprana y persiste hasta el final de la vida. No es considerado un trastorno progresivo, sin embargo, la clínica cambia con el tiempo a medida que el cerebro madura y se desarrolla (Elena Beani et al, 2019) (Benedetto et al., 2023).

Se caracteriza por una alteración de la coordinación, así como de la regulación del tono muscular, debido a la debilidad, las anomalías del control motor y la espasticidad, es decir, la rigidez y tensión de los músculos.

Puede ir acompañado de otras disfunciones a nivel sensorial, cognitivo, de comunicación, de percepción, del comportamiento, la comunicación, de epilepsia y trastornos músculo esquelético secundarios (Benedetto et al., 2023) (Menor-Rodríguez et al., 2021).

1.2 Etiología

La etiología de la parálisis cerebral es diversa y se puede deber a multitud de factores. Dichos factores se pueden clasificar en cuatro grandes grupos de alteraciones: antes de la concepción, relacionado con la salud y las condiciones de vida de la madre en términos generales; prenatales, relacionados con el curso del embarazo; perinatales, durante el parto; y neonatales e infantil.

En la Figura 1 podemos visualizar los factores de riesgos asociados a cada uno de los periodos críticos en los que se puede llegar a producir esta patología. Ponemos especial atención al periodo prenatal, siendo el origen del 70-80% de los casos de parálisis cerebral.

Durante el periodo prenatal, puede llegar a producirse la hipoxia neonatal, consecuencia de anomalías del desarrollo, anomalías metabólicas, defectos autoinmunes y de la coagulación, infecciones, traumatismos o combinaciones de estos (Vela & Ruiz, 2014).

También destaca como factor de riesgo el embarazo múltiple, elevando el riesgo cuando se produce un parto de mellizos o trillizos.

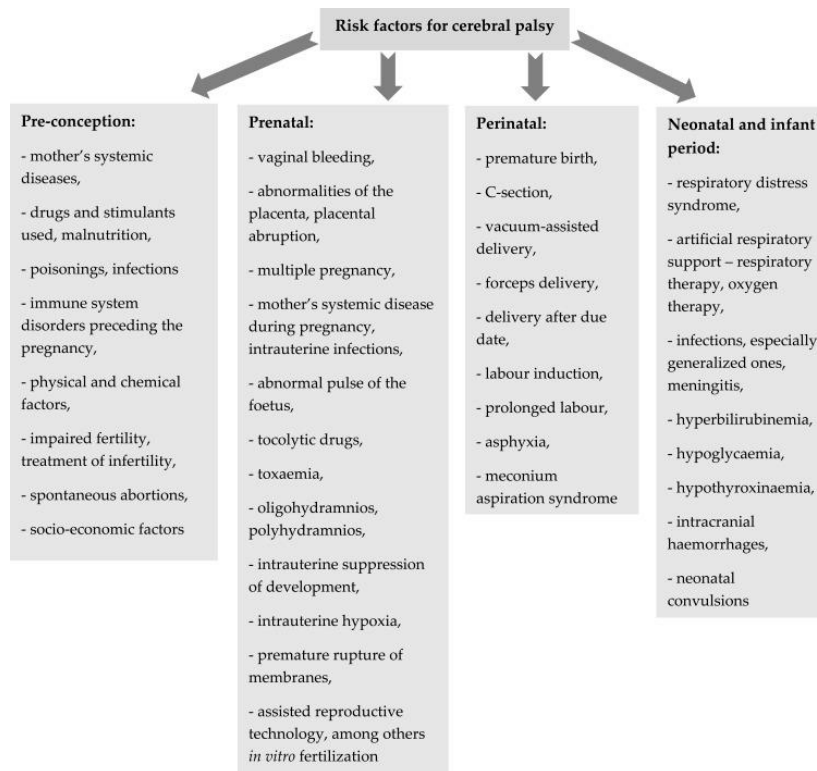


Figura 1. Factores de riesgo potenciales de la parálisis cerebral (Sadowska et al., 2020)

Otro de los principales factores de riesgo de incidencia de parálisis cerebral, es la prematuridad. Se ha llegado a una correlación entre la frecuencia y la gravedad de los trastornos del neurodesarrollo con la duración del embarazo, llegando a la conclusión de que, cuanto más corto es el embarazo, más graves son los trastornos. Otro factor de riesgo determinante es el peso al nacer en bebés prematuros.

1.3 Clasificación

Existen numerosas clasificaciones de la parálisis cerebral pediátrica dependiendo del enfoque, pudiéndose abordarse según las partes del cuerpo afectadas, cuan grave es la sintomatología o por el trastorno neuromotor presente. Las clasificaciones más empleadas son las propuestas por Ingram (1955) y Hagberg (1976) (Elena Beani et al, 2019).

La clasificación de Ingram lo aborda desde el tipo de síndrome neurológico y su localización, así como, según la gravedad de los síntomas. Diferenciando entre diplejía, hemiplejía, hemiplejía bilateral (tetraplejía), ataxia, discinesia y tipos mixtos (Sadowska et al., 2020).

Según la parte del cuerpo afectada, distinguimos monoparesia si solo afecta a un miembro, generalmente el brazo; diplejía, si afecta a ambas piernas; hemiplejía cuando tienen afectado un lado del cuerpo con mayor compromiso de la extremidad superior y tetraparesia, afectación de los 4 miembros (Argüelles, 2012; ASPACE, 2017).

En todos los tipos, los trastornos del movimiento pueden ser leves, moderados o severos.

Mientras que la clasificación de Hagberg, tiene en cuenta: síndromes espásticos, síndromes extrapiramidales (disquinéticos) y ataxia (Sadowska et al., 2020).

La parálisis cerebral espástica, es el tipo más común, afectando al 80% de las personas, que la padecen. Tienen dificultad para controlar los músculos, tendiendo a estirarse y debilitarse.

Otro tipo es la parálisis cerebral discinética, se relaciona con factores perinatales, caracterizado por una fluctuación y cambios bruscos del tono muscular, presencia de movimientos involuntarios y persistencia de reflejos arcaicos, es decir, reflejos de recién nacido.

Podemos diferenciar otro tipo de parálisis cerebral, la atáxica, cuyos síntomas son la hipotonía, la ataxia, la disimetría y la incoordinación (Argüelles, 2012).

Dentro de la parálisis cerebral espástica, el tipo más frecuente es la hemipléjica comprendiendo entre el 30-40% de los niños con parálisis cerebral. Provoca trastornos de movimiento de las extremidades superiores, teniendo una prevalencia entre el 57-83%. Disminuyendo la capacidad unimanual, a su vez que disminuye el rendimiento bimanual, repercutiendo en las actividades de la vida diaria (Elena Beani et al, 2019) (Cacioppo et al., 2020).

1.4 Escalas para la evaluación del miembro superior

Para realizar una valoración de los trastornos motores de los pacientes con parálisis cerebral, se pueden aplicar varias escalas. Una de las más comunes y fáciles de usar es el Sistema de la Función Motora Gruesa (GMFCS), se basa en evaluar la independencia del niño al realizar funciones motoras básicas, como puede ser caminar o moverse con la ayuda de equipos auxiliares. Sin embargo, carece de evaluación de la destreza manual.

Para evaluar la función de las extremidades superiores se puede emplear herramientas como es el The Manual Ability Classification System (MACS), The Assisting Hand Assessment (AHA) o The Melbourne Assessment (MA) (Sadowska et al., 2020).

1.4.1 The Manual Ability Classification System (MACS)

The Manual Ability Classification System describe como los niños, de 4 a 18 años, con parálisis cerebral usan sus manos para manejar objetos en las actividades de la vida diaria.

Esta clasificación se divide en 5 niveles, basados en la capacidad autoiniciada de los niños para manejar objetos y su necesidad de asistencia o adaptación para realizar las actividades en la vida cotidiana.

Se distingue el nivel I, en el que maneja los objetos con facilidad y con éxito; en el nivel II, el paciente maneja la mayoría de los objetos con cierta calidad y/o velocidad reducida.

Dentro del nivel III, el niño maneja los objetos con dificultad y requiere ayuda para modificar las actividades. En el nivel IV, maneja una selección limitada de objetos fáciles de administrar y siempre requiere ayuda de otros.

Por último, el nivel V, el niño no es capaz de manipular objetos ni de realizar ni siquiera acciones sencillas con las manos (Alliance, 2018).

Es muy útil para comprender las situaciones en las que el niño es independiente y hasta qué punto necesita apoyo o adaptaciones (MACS, 2021).

1.4.2 The Assisting Hand Assessment (AHA)

The Assisting Hand Assessment es una escala que valora tareas bimanuales en niños que presentan una discapacidad unilateral, por ejemplo, parálisis cerebral hemipléjica. Se realiza en pacientes pediátricos entre los 18 meses y 12 años.

Su objetivo es medir y describir el uso efectivo de la mano afectada paralelamente con la mano sana, durante la realización de las actividades bimanuales en un entorno de juego natural. En el transcurso de la prueba, se valora el uso espontáneo y la calidad de la mano asistente y no la capacidad máxima de la mano afectada.

Los dominios de la escala The Assisting Hand Assessment son los siguientes: uso general del brazo, presión-liberación, ajustes motrices finos, coordinación y ritmo. Se valoran dando una serie de juguetes, predeterminados por la escala.

Esta escala se está utilizando actualmente para la valoración de la efectividad de terapias del miembro superior afectado en la parálisis cerebral (Casares, 2015).

1.4.3 The Melbourne Assessment (MA)

The Melbourne Assessment es una prueba de evaluación de alta fiabilidad de la función unilateral de las extremidades superiores en niños con deficiencias neurológicas de 2.5 a 15 años.

Durante la prueba se toma como referencia cuatro elementos de la calidad del movimiento: (1) rango de movimiento, (2) precisión de alcance y colocación, (3) destreza de agarre, liberación y manipulación y (4) fluidez de movimiento (Park et al., 2021).

La prueba completa consta de 14 tareas que suponen al niño alcanzar, agarrar, soltar y manipular objetos simples. Durante el proceso se graba en vídeo para su posterior calificación por el terapeuta o médico rehabilitador (Park et al., 2021).

La puntuación se completa a través de los 30 ítems de puntuación utilizando un sistema con escala de 3, 4 o 5 puntos, donde las puntuaciones más bajas representan un peor deterioro. Las puntuaciones de los ítems relacionadas con cada elemento de movimiento medido se suman dentro de la subescala correspondiente. El puntaje final de un niño, se informa como cuatro puntuajes separados, uno para cada elemento de la calidad del movimiento medido (Bourke-Taylor, 2003).

Para los niños con afectación bilateral, se puede evaluar una de las extremidades superiores. Si se van a evaluar ambas, cada miembro se evalúa y puntúa por separado (Park et al., 2021) (Randall et al., 2014).

2. SENSORES INERCIALES: XSENS AWINDA

2.1 Introducción a los sensores inerciales

Los sensores inerciales o *Inertial Measurement Unit* (IMU), son pequeños dispositivos capaces de medir la aceleración lineal y la velocidad angular, utilizados para cuantificar y valorar de forma objetiva el movimiento corporal humano.

Estos instrumentos están compuestos por una masa, un par de resortes y un capacitor, que permite calcular la aceleración lineal y velocidad angular de los movimientos en tres ejes preestablecidos (x, y, z) de manera precisa.

Por un lado, los acelerómetros captan dicha información, y a partir de ella, son capaces de calcular la velocidad en relación con la ejecución del movimiento, para una posterior transformación a aceleración.

Por otro lado, los giroscopios se encargan de los cambios posicionales, ofreciendo la posibilidad de medir la ubicación de una estructura corporal en relación con un punto fijo (Castellanos-Ruíz et al., 2020).

Los IMU se transforman en una herramienta novedosa en desarrollos tecnológicos, sobresaliendo en la captura y análisis de movimiento para su posterior aplicación en la evaluación del movimiento corporal humano.

Se han desarrollado sensores más completos que incluyen acelerómetros, giroscopios y magnetómetros que permiten conocer la orientación 3D de dichos sensores con un alto nivel de precisión (Li et al., 2017) (Palermo et al., 2022).

2.2 Introducción XSens Awinda

El sistema XSens Awinda está compuesto por 17 unidades de medición o sensores inerciales (IMU), siendo rastreadores inalámbricos, sincronizados a alta precisión.

Estos, son fácilmente transportables gracias al maletín donde vienen colocados y facilitan su correcto almacenaje y desplazamiento. Podemos visualizar el maletín mencionado en la Figura 2.



Figura 2. Maletín del Sistema XSens Awinda junto con sus sensores inerciales (Movella, 2022)

Se encuentran sujetos al cuerpo del sujeto con una camiseta y un juego de correas. Además, cuenta con una unidad con antena, el dongle receptor, que se conecta al ordenador portátil que posee el *software* de este sistema.

Cada sensor, mostrado en la Figura 3, mide directamente la orientación y no la posición, por lo que, la precisión de las posiciones de los sensores no es muy relevante, consiguiendo eliminar el sesgo del operador cuando se realizan mediciones en días diferentes. Cuando se realiza la calibración en el *software*, la orientación de cada sensor se alineará con la orientación de cada segmento del cuerpo (Movella, 2022) (Movella, 2021).



Figura 3. Sensores inerciales del Sistema XSens Awinda

2.3 Colocación sensores

Nos centraremos en la colocación del sistema XSens Awinda en la extremidad superior, ya que, este estudio se centra en ello. Sin embargo, se puede utilizar en todo el cuerpo humano colocándolo por las extremidades inferiores.

En primer lugar, se colocará un sensor en la cabeza del sujeto, este se introducirá en una diadema deportiva alrededor de su cabeza, priorizando la estabilidad del sensor, como observamos Figura 4.



Figura 4. Sensor de la cabeza del Sistema XSens Awinda (Movella, 2022)

En los hombros, se situarán dos sensores de forma simétrica en dos parches en la parte superior de la camiseta, se colocarán de forma horizontal, de forma que la totalidad del sensor entre en contacto con el parche de velcro, tal y como se muestra en la Figura 5.



Figura 5. Sensores de los hombros del Sistema XSens Awinda

En el brazo se colocará un sensor en la parte superior lateral del brazo, entre los grupos musculares del bíceps y el tríceps. Así como, otro sensor en el antebrazo, cerca de la muñeca, proporcionando información sobre las rotaciones de pronación y supinación.

En la mano, cuenta con un guante especial para cada mano, este dispone de un bolsillo, para mejorar el ajuste, donde se colocará cada sensor de forma vertical.

Además, en la parte delantera de la camiseta cuenta con un bolsillo donde se coloca el sensor del esternón, mejorando su fijación.

Los tres sensores del brazo, anteriormente mencionados, se colocan en ambas extremidades superiores de forma simétrica, como podemos visualizar en la Figura 6.



Figura 6. Sensores de las extremidades superiores del Sistema XSens Awinda

Por último, para capturar el movimiento de la pelvis, se dispone de una correa a la altura de la espina ilíaca anterosuperior, cerciorándonos de estar colocado en la pelvis y no en la parte alta de la cintura. Es importante colocar el sensor orientado verticalmente, al ser el único de los sensores con una orientación especificada (Movella, 2022).

2.4 Calibración sensores

La calibración de los sensores XSens Awinda se realiza una vez que han sido colocados todos los sensores en sus respectivas localizaciones, asegurándolos con las cintas y en los lugares habilitados dentro de la camiseta.

El propio programa XSens Awinda va guiando en todo el proceso de calibración, indicando las instrucciones de manera oral. Antes de comenzar el calibrado, muestra unos ejemplos de cómo se debe posicionar el sujeto, tal y como se muestra en la Figura 7.

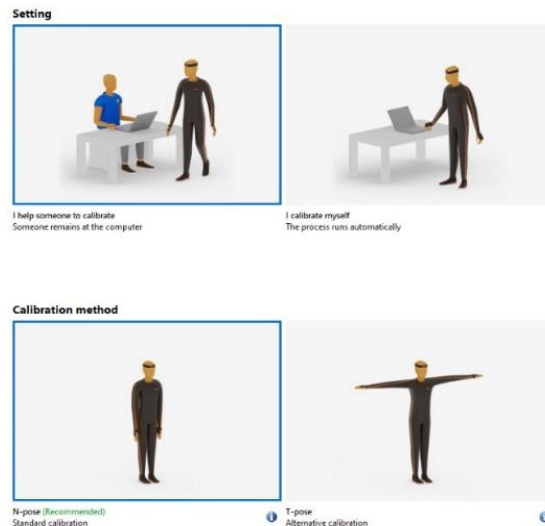


Figura 7. Posición neutral de calibración del Sistema XSens Awinda

La posición que debe adoptar el participante consiste en la *N-pose*, en la que debe situarse de pie con las piernas juntas, los brazos pegados al tronco y con la cabeza mirando al frente. Una vez que se inicia el procedimiento el sujeto debe aguantar durante 3 segundos dicha pose, hasta que el programa lo indique.

A continuación, el sujeto debe caminar relajado y evitando movimientos bruscos durante 10 – 15 segundos. Posteriormente, el programa solicita que el sujeto vuelva a la *N-pose*, durante lo que queda de procedimiento. Al finalizar, se indica si se ha realizado correctamente la calibración.

2.5 Grabación

Previamente a la grabación, se debe indicar ciertas configuraciones para determinar el modo de grabación. En primer lugar, la configuración del traje, pudiendo elegir entre distintas opciones: todo el cuerpo, la parte superior del cuerpo, la parte inferior o la parte inferior junto con el esternón.

También requiere medidas obligatorias del sujeto, como la altura y el tamaño del pie en centímetros. Se pueden configurar manualmente otras medidas como la altura de los hombros o de la cadera, así como, el ancho de los hombros y la cadera. Si no se introducen, el programa realiza una estimación de ellos, para que el modelo se asemeje al participante.

Una vez realizada la grabación, se pueden acceder a las grabaciones y volver a visualizarlas. No obstante, no se puede acceder a los datos hasta que no se paga el acceso a la nube.

3. VISIÓN ARTIFICIAL: OPENCAP

3.1 Introducción a la visión artificial

El ser humano percibe las imágenes a través de la retina, obteniendo información visual del entorno, es decir, se comporta como una cámara fotográfica sensible a la luz. Dentro de él se encuentran los conos y bastones, estructuras sensibles a los tres colores básicos del *RGB*. Tras la captura de información por la retina, está se transmite al cerebro por medio del nervio óptico (Sanabria S. & Archila D., 2011).

La visión artificial trata de imitar la forma en la que el ojo reconoce, analiza y entiende una escena y sus componentes. Por lo que, definimos la visión artificial como la ciencia de programar un computador para procesar imágenes o vídeos e incluso llegar a entenderlos, intentando imitar la capacidad de los seres humanos para ver una imagen, entenderla y actuar en consecuencia.

Los componentes de un sistema de visión artificial incluyen la iluminación, las lentes, el sensor de imagen, el procesamiento de visión y comunicaciones. La iluminación permite que las lentes capturen mejor la imagen y la presenta al sensor en forma de luz, que la convierte en una imagen digital, que se envía al procesador para su posterior análisis. El procesamiento consiste en una serie de algoritmos que revisan y extraen la información.

Existen 3 categorías de sistemas de visión artificial: 1D, 2D y 3D. No obstante, nos centraremos en el subtipo de sistemas 3D. Estos sistemas emplean múltiples cámaras, dispuestas de tal forma que permiten conocer la ubicación de los objetos de interés por triangulación (COGNEX, 2018).

3.2 Introducción OpenCap

OpenCap es un *software* de código abierto, desarrollado por la Universidad de Stanford en el departamento de biomecánica neuromuscular.

Su funcionalidad consiste en estimar la cinemática 3D y la dinámica del movimiento humano, combinando la visión por ordenador, el *Deep Learning* y la simulación musculoesquelética, a partir de vídeos capturados con dos o más smartphones de forma sincronizada (Uhlrich et al., 2022a) (Uhlrich et al., 2022b).

3.3 Conceptos teóricos acerca del funcionamiento

El *software* OpenCap procesa las grabaciones obtenidas mediante dos algoritmos de detección de pose: OpenPose y HRNet. Estos algoritmos fueron seleccionados por sus buenos rendimientos, gracias a la inclusión de puntos clave en los pies del sujeto.

Con estos algoritmos se pueden generar entradas para modelos de aprendizaje automático y así, lograr una estimación de las medidas cinemáticas y cinéticas del participante (Uhlrich et al., 2022a).

Primeramente, se realiza un análisis cinemático 2D, para cada *frame* de los vídeos grabados, los algoritmos devuelven la posición bidimensional (2D) de los puntos clave del participante. Se obtienen veinte puntos clave de partes del cuerpo como el cuello, la parte media de la cadera, los hombros, las caderas, las rodillas, los tobillos, los talones, los dedos gordos y pequeños de los pies y las muñecas.

Además, dichos algoritmos procesan las posiciones de los puntos clave 2D cuando se producen oclusiones durante la grabación y los sincroniza en el tiempo, utilizando correlaciones cruzadas de velocidades de dichos puntos clave.

Posteriormente, dichos puntos 2D se triangulan, calculando los puntos clave 3D, usando un algoritmo de transformación lineal directa y se convierten en un conjunto de cuarenta y tres marcadores anatómicos 3D, obteniendo un modelo más completo.

A continuación, se calcula la cinemática 3D a partir de las trayectorias de los marcadores mediante cinemática inversa y un modelo musculoesquelético con restricciones biomecánicas. Finalmente, para las medidas cinéticas se estiman mediante simulaciones dinámicas, que siguen la cinemática 3D (Uhlrich et al., 2022b).

Tras el proceso de calibración, OpenCap graba al participante en una postura neutra de pie que, junto con las posiciones de los marcadores anatómicos, escala un modelo musculoesquelético de la antropometría del participante, gracias a la herramienta Scale de OpenSim (Uhlrich et al., 2022a).

3.4 Calibración

Antes de realizar la calibración de los dispositivos, se colocarán los iPads, de tal manera que su configuración será de 30-45° respecto a la línea en la que se ubicará el sujeto. Evitando en todo momento las vistas sagitales puras, al no funcionar bien debido a la oclusión de las extremidades.

Se recomienda observar al participante realizar la tarea de interés en cada cámara antes de la calibración, para asegurarnos de que permanece dentro del campo de visión. Respecto al entorno, se debe realizar en un lugar bien iluminado (Uhlrich et al., 2022a).

Una vez las cámaras se encuentran sincronizadas, el proceso de calibración se realiza a través de un tablero de ajedrez, que se debe visualizar en las cámaras de todos los dispositivos.

Dicho tablero se imprimirá en un folio de tamaño A4, con cuadrados de 35 mm como mínimo, con un borde blanco en todos los lados. Respecto al número de columnas y filas del tablero se especificará en la aplicación web, introduciendo el número de esquinas negro – negro, en vez del número de cuadrados de las columnas o filas.

En la página oficial de OpenCap, se facilita un modelo con las dimensiones recomendadas, listo para imprimirlo, podemos visualizar el ejemplo en la Figura 8 (Uhlrich & Falisse, s. f.).

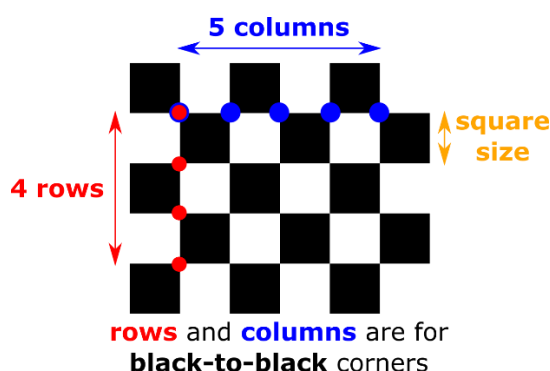


Figura 8. Tablero de calibración de OpenCap

El tablero debe encontrarse a menos de 5 metros de las cámaras y ser visible en cada una de ellas, encontrándose en el centro del sujeto que se quiere capturar, lo más cerca de las cámaras, siendo visible por todas, en la Figura 9 se puede visualizar la correcta colocación del tablero.

Además, la línea que sale del tablero debe bisecar el arco de los iPads y no apuntar directamente a ellos, evitándolo en todo momento.

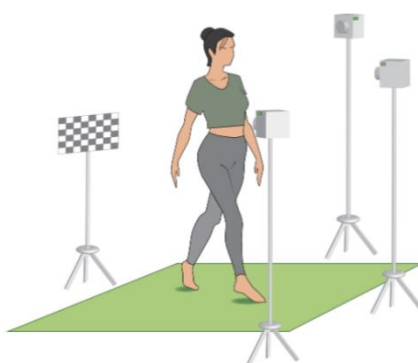


Figura 9. Disposición del tablero y las cámaras

Asimismo, el tablero de ajedrez se debe posicionar de forma perpendicular al suelo con su borde más largo también paralelo al suelo (Uhlrich & Falisse, s. f.).

Después de realizar la calibración de las cámaras, se grabará al sujeto en una postura neutra de pie. El participante se situará de pie con el tronco recto, las piernas abiertas a la misma altura que el ancho de las caderas, y los brazos abiertos, formando un triángulo con el tronco y las manos con las palmas mirando hacia el cuerpo, tal y como se muestra en la Figura 10.

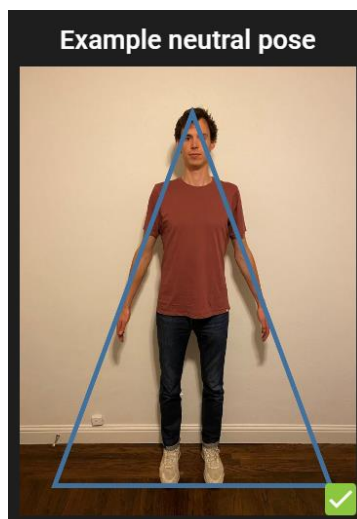


Figura 10. Posición neutral de calibración de OpenCap

3.5 Grabación

En primer lugar, para realizar una grabación con OpenCap necesitamos al menos dos dispositivos iOS, ya sean iPhone, iPad o iPod, y durante la grabación estarán sujetos por trípodes.

Además, requiere un tablero de calibración, que desde la página web oficial se facilita y se puede imprimir desde una impresora estándar. Por último, se necesita un dispositivo conectado a la red para ejecutar la aplicación web de OpenCap, por ejemplo, un portátil.

En los dispositivos iOS que se van a emplear, se necesitan dos aplicaciones gratuitas, que se encuentran en el App Store: TestFlight y OpenCap.

El proceso de grabación se realizará mediante los dos iPads sujetos cada uno a un trípode y conectados a una red *Wifi*, dispuestos a cierta distancia, adaptado a las características y dimensiones de la sala de grabación.

Los vídeos se graban con una resolución de 720x1280 *píxeles*, una frecuencia de imagen de 60 Hz, y con la distancia de enfoque de la cámara ajustada a un valor fijo.

Para las grabaciones se utilizará la cámara situada en la parte trasera del dispositivo, colocadas de tal manera que se eviten las oclusiones de segmentos, así como, evitando que aparezcan personas en segundo plano. Además, las dos cámaras deben ver todos los segmentos corporales en todo el momento durante el movimiento.

Para sincronizar las dos cámaras con el ordenador, al iniciar sesión en la aplicación de OpenCap, se dispondrá de un código QR, como el de la Figura 11, que se escaneará en las cámaras de los iPads.



Figura 11. Código QR para sincronizar las cámaras de OpenCap

Una vez que se ha escaneado el código se procede a la calibración con el tablero de ajedrez. Tras esto, el sujeto se coloca delante de las cámaras y se graba en posición de la pose neutral.

Una vez que se han conseguido realizar ambos procesos correctamente, se llevaran a cabo la grabación de los movimientos para evaluar la funcionalidad de la extremidad superior.

3.6 Comparación con otros métodos

Existen otros métodos para capturar el movimiento como puede ser los métodos que emplean marcadores, realizados en un laboratorio especializado. Podemos analizar las prestaciones de ambos métodos, como se ilustra en la Figura 12.

Sin embargo, requiere de varios días de medición, así como, un equipo de expertos que la lleven a cabo. Con OpenCap, nos permite hacer un estudio del movimiento musculoesquelético en menos de 10 minutos, desde cualquier lugar con acceso a internet, requiriendo un mínimo de dos dispositivos iOS (Uhlrich et al., 2022a) (Uhlrich et al., 2022b).

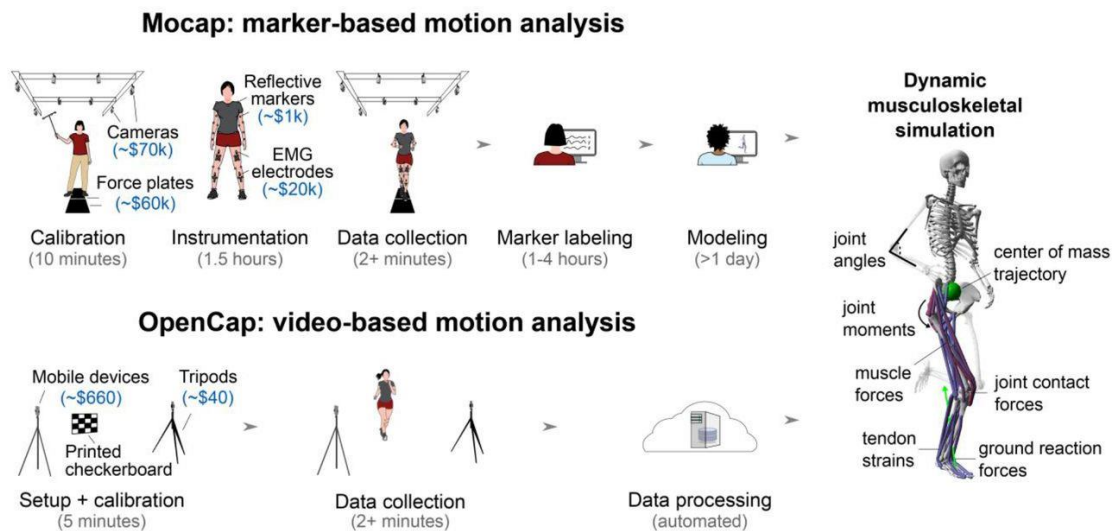


Figura 12. Captura de movimiento basada en marcadores (MOCAP) comparado con análisis basado en vídeo (OpenCap) (Uhlrich et al., 2022b).

Dentro del análisis del movimiento basado en marcadores destaca el Sistema de Análisis Vicon de fotometría 3D. Consiste en la utilización de marcadores reflectantes en determinados puntos articulares, que son capaces de capturar la luz de unas cámaras infrarrojas.

Gracias a dichos puntos se forma un modelo biomecánico de *Plug in Gait*, es decir, tiene en cuenta un número mínimo de marcadores, junto con una serie de medidas del cuerpo humano. A través del modelo, es posible reconocer la figura humana, con sus segmentos y medidas reales del sujeto de estudio y tras el procesamiento de los datos, se obtiene datos reales sobre los movimientos articulares.

Proporciona información de la cinemática, obteniendo los ángulos articulares de todas las articulaciones en los 3 ejes de dirección (x, y, z), consiguiendo tener datos en todos los planos, flexo-extensión, abducción-aducción y rotaciones. Además, podemos adquirir las aceleraciones angulares, velocidades angulares y aceleraciones lineales, así como las trayectorias de un punto y las distancias de desplazamiento en cualquier eje de dirección (Bernad, 2021).

Este sistema presenta grandes ventajas respecto a otros sistemas, no obstante, su precio es elevado respecto a la oferta del mercado.

4. DEEP LEARNING

4.1 Introducción al Deep Learning

La Inteligencia Artificial es una rama de las ciencias de la computación, que incluye conceptos transversales relacionados con la lógica y el aprendizaje. Se diseñan herramientas informáticas, que simulan procesos de la inteligencia humana que incluyen el aprendizaje, el razonamiento y la autocorrección. Para ello, se emplean una serie de algoritmos lógicos suficientemente entrenados, a partir de los cuales las máquinas son capaces de tomar decisiones para casos concretos a partir de normas generales (Ávila-Tomás et al., 2021).

Durante la última década, la inteligencia artificial se ha convertido en un tema en auge tanto dentro de la comunidad científica como fuera. La inteligencia artificial cae dentro del ámbito de la ciencia de datos e incluye la programación clásica y el aprendizaje automático (Choi et al., 2020).

Mientras que el aprendizaje humano es progresivo y se desarrolla a partir de la exposición a situaciones de forma consecutiva, la adquisición de estos datos por parte de una máquina es mucho más rápida (Ávila-Tomás et al., 2021).

El aprendizaje automático es una rama en evolución de los algoritmos computacionales que están diseñados para emular la inteligencia humana aprendiendo del entorno. Las técnicas basadas en aprendizaje automático se han aplicado en numerosos campos, desde el reconocimiento de patrones, visión por computadora, entretenimiento y la biología computacional hasta las aplicaciones biomédicas y médicas (El Naqa & Murphy, 2015).

El *Deep Learning* es un tipo particular de Machine Learning, como podemos ver en la Figura 13, una técnica que permite a los ordenadores el poder de mejorar de manera gradual con datos y experiencia. Obtiene buenos resultados, gracias a que representa problemas del mundo real como modelos computacionales, basando los conceptos más complejos en función de aquellos más simples. Es decir, descubre estructuras intrínsecas en grandes conjuntos de datos mediante el uso del algoritmo de *Backpropagation* (Lecun et al., s. f.).

El reconocimiento de la actividad humana a través de señales de sensores con acelerómetros y giroscopio ha avanzado con el uso de técnicas, así como, de dispositivos móviles inteligentes, redes de comunicación inalámbrica y aprendizaje automático.

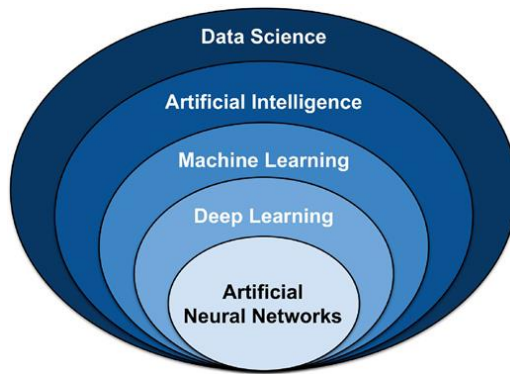


Figura 13. Técnicas del análisis de datos (Choi et al., 2020)

4.2 Redes neuronales

Las redes neuronales o redes neuronales artificiales se engloban dentro de los algoritmos de *Deep Learning*, son un modelo matemático inspirado en la estructura del cerebro humano, reproduciendo la forma en la que las neuronas biológicas transmiten la información de unas a otras.

Se componen de capas de nodos, que contienen una capa de entrada, una o más capas ocultas, y una capa de salida. Cada nodo, o neurona artificial, se conecta a otra y tiene un peso y un umbral asociados. Si la salida de cualquier nodo individual está por encima del valor umbral, dicho nodo se activa, enviando los datos a la siguiente capa de la red (IBM, 2021).

Las redes se entrenan con unos datos, con los que aprenden y mejoran su precisión con el tiempo, el conjunto de entrenamiento. Además, existen dos grupos más, el conjunto de validación, que sirve para parar el entrenamiento cuando la red comienza a sufrir *overfitting* y, por último, el conjunto de *test*, el cual evalúa el rendimiento de la red, al ser datos nunca visto por la red. Consiguen que tareas de clasificación que pueden llegar a durar horas, se realicen en cuestión de minutos.

Las principales aplicaciones de las redes neuronales se han centrado en la predicción de la postura del modelo humanos y la predicción de movimientos robóticos y dinámicos (Bataineh et al., 2016).

4.3 Redes convolucionales

Entre las arquitecturas que destacan dentro de las redes neuronales, sobresalen las redes neuronales convolucionales. Siendo una de las técnicas más poderosas para el reconocimiento de la actividad humana.

Son de gran utilidad cuando se emplean numerosos sensores para el análisis del movimiento, ya que entre ellos hay una gran dependencia espacial, así como, entre los ejes de los acelerómetros y giroscopios que llevan incorporados.

La red neuronal convolucional comprende una capa de entrada, una o más capas convolucionales y de agrupación, seguidas de una o más capas, completamente conectadas.

La operación de convolución se aplica a los nodos de entrada o a los nodos ocultos en la capa anterior, calculando una suma ponderada de los valores de los nodos dentro del *kernel* de convolución, seguido de agregar un sesgo y pasarlo a través de una función de activación para formar mapas de características (Ha & Choi, 2016)

4.4 Autoencoders

Un autoencoder es un tipo específico de red neuronal con numerosas capas ocultas, cuya finalidad es reducir la dimensionalidad de la base de datos, reconstruyendo los datos entrada.

Dicha red utiliza los datos de salida de las capas ocultas, que son más pequeños que los datos de entrada, para reconstruir los datos de entrada y compararlos con los originales de entrada. Además, estas redes se emplean en el aprendizaje no supervisado utilizando el método del descenso de gradiente, como la retropropagación (Tan & Eswaran, 2008).

Un autoencoder típico está formado por dos partes principales: el codificador y el decodificador, como podemos visualizar en la Figura 14. La red “codificadora” transforma los datos de entrada, reduciendo su dimensión, mientras que la red “decodificadora” reconstruye los datos originales a partir de los datos que proporciona el “codificador” (Tan & Eswaran, 2008).

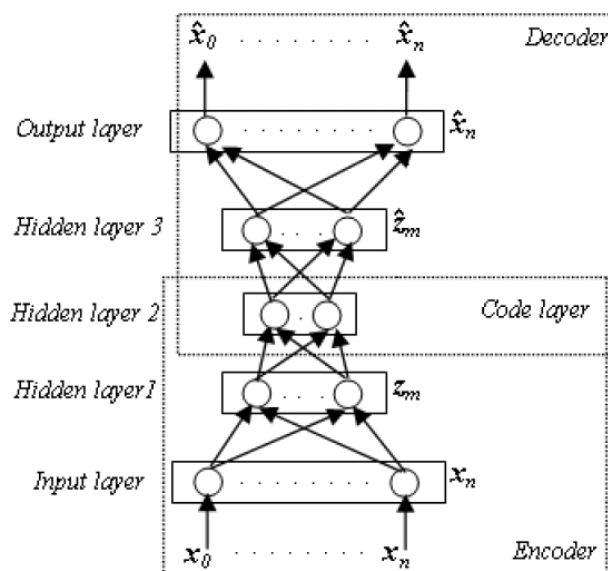


Figura 14. Estructura de un autoencoder (Pathirage et al., 2018)

Consta de dos etapas durante el entrenamiento: el pre-entrenamiento y el ajuste fino. El pre-entrenamiento se realiza capa por capa, empleando múltiples autocodificadores simples para inicializar los pesos de cada una.

La segunda etapa, el ajuste fino, consiste en optimizar las múltiples capas de la red respecto al objetivo de la función final, utiliza la retropropagación para ajustar toda la red con los datos (Pathirage et al., 2018).

4.5 Técnicas de regularización

En *Deep Learning* es muy común que, al construir los modelos y entrenarlos, se produzca *overfitting* o *underfitting*. Esto es debido a los numerosos parámetros que puede contener una red, es sencillo que pueda sobre ajustar, en estos casos, las técnicas de regularización tratan de solventar el problema, mejorando el rendimiento del modelo.

El fenómeno de *overfitting* o sobre ajuste, se produce cuando el modelo aprende muy bien los datos de entrenamiento, sin embargo, no es capaz de generalizar a datos nunca vistos. Asimismo, el *underfitting*, es el proceso contrario, el modelo no se ajusta bien a los datos de entrenamiento y predice de manera insatisfactoria al no haber sido capaz de identificar patrones (Edward Joseph Velo Fuentes, s. f.).

Las técnicas de regularización son estrategias que se aplican a las redes convolucionales y dependiendo de lo que se busque y el objetivo a tratar se aplican unas u otras.

En primer lugar, destacamos el método *DropOut*, esta estrategia trata de evitar el *overfitting*, en base a la desactivación de forma aleatoria y temporal de neuronas pertenecientes a las capas ocultas dentro del modelo. Las neuronas desactivadas no se incluyen a la hora de realizar los cálculos de los pesos, y obliga a las neuronas vecinas a no depender de las neuronas desactivadas. Es una forma de reducir el número de parámetros estimados, así como, de la complejidad de la red.

También sobresale el procedimiento de *Batch Normalization*, el cual se fundamenta en la adición de un paso extra, entre las neuronas y la función de activación, que consiste en la normalización de las activaciones de salida. Se realiza entre mini *batches* en vez de al conjunto de datos completo, acelerando el entrenamiento de la red.

Distinguimos otros métodos como la regularización L1 o regularización Lasso y la regularización L2 o Ridge. Ambos son procedimientos muy similares, se basan en la actualización de la función de pérdidas, añadiendo un nuevo parámetro.

En el caso de la regularización L2, es la suma de las normas al cuadrado de la matriz de pesos. Para la regularización L1, el término es el sumatorio de los valores absolutos de los pesos, reduciendo la influencia de los pesos menos importantes.

Early Stopping es una técnica simple y efectiva, que consiste en detener el ajuste del modelo en un punto donde la función de pérdidas del set de validación empieza a incrementarse. Se puede utilizar de dos maneras distintas, una vez se detecta la

época óptima, se reinicia el modelo con todos los datos del set de entrenamiento, sin set de validación y se detiene el entrenamiento en dicha época.

Otra forma es entrenar el modelo con un set de validación, se detiene el ajuste cuando se alcanza la época y se continúa el entrenamiento con los nuevos datos, hasta que se alcance el nuevo mínimo error que supere al anterior (Edward Joseph Velo Fuentes, s. f.).

Finalmente, cabe mencionar la técnica de *Reduce Loss on Plateau*, que reduce la tasa de aprendizaje una vez que se estanca la métrica de interés. Se define un número de épocas que, si no se ve ninguna mejora, la tasa de aprendizaje se reduce.

4.6 Métricas de rendimiento en redes neuronales

Las métricas de rendimiento son fundamentales a la hora de evaluar la calidad de los algoritmos de *Deep Learning*, permiten seleccionar cual es el mejor según el objetivo de la investigación. Dentro de las métricas de rendimiento podemos diferenciar numerosas medidas (Borja-Robalino et al., 2020).

Una de ellas es la matriz de confusión, es una representación matricial de los resultados de las predicciones de un conjunto de datos que no ha sido utilizado en el proceso de entrenamiento. Para los problemas multiclase, es un método muy eficiente y fácil de entender para sacar conclusiones sobre el rendimiento del algoritmo. Se debe elegir una métrica que es la que se representa dentro de la matriz de confusión.

Destacamos la *accuracy* o exactitud, como la métrica más utilizada, tanto para casos dicotómicos como multiclase. Mide el porcentaje de casos que el modelo ha acertado. Para modelos en los que los datos están desbalanceados es útil emplear otras métricas, dado que esta puede generar confusión (Borja-Robalino et al., 2020; Martínez Heras, 2019).

Otra de las métricas es *Recall* o sensibilidad, dicha medida permite conocer la proporción de casos positivos correctamente clasificados, es decir, verdaderos positivos.

Paralelamente a la sensibilidad, se estudia la especificidad, como la proporción de casos negativos bien clasificados, es decir, verdaderos negativos.

La métrica de precisión consiste en calcular el número de verdadero positivos que son realmente positivos, en comparación con el número total de valores positivos predichos (Martínez Heras, 2019).

La medida de *F1-score* combina ambas métricas, *Recall* y *Accuracy*, en una media armónica de ellas, siendo un indicador general de la precisión, en problemas de desbalanceo de clases. Sobre todo, se tiene en cuenta en conjuntos de datos

desbalanceados, con una gran presencia de verdaderos negativos (Borja-Robalino et al., 2020).

Dos métricas que suelen emplearse son la tasa de falsos negativos y tasa de descubrimiento falso. La tasa de falsos negativos (FNR) es la proporción de casos positivos que la red clasifica como negativo y la tasa de descubrimiento falso (FDR), es el número de falsos positivos que se espera obtener.

Por último, otro estadístico muy empleado, es el valor predictivo positivo, siendo la probabilidad de que un sujeto clasificado como negativo, verdaderamente no este afectado.

5. MATERIALES Y MÉTODOS

5.1 Bases de datos

Durante este proyecto se adquirió una base de datos de distintos movimientos del miembro superior, tanto unimanuales como bimanuales. Sin embargo, nos centraremos más en los movimientos bimanuales, al estar más presentes en la vida diaria de los pacientes.

En este proceso se utilizaron tanto sensores vestibulares como tecnologías de captura de vídeo con el objetivo de capturar el movimiento. Participaron en la captura adultos y niños voluntarios sanos y niños diagnosticados con parálisis cerebral.

Las grabaciones se llevaron a cabo en la Escuela Técnica Superior de Ingenieros de Telecomunicación con los adultos y en el Hospital Universitario Río Hortega de Valladolid con niños sanos y con parálisis cerebral.

5.2 Materiales

En este proyecto hemos utilizado las instalaciones de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid, en concreto, la sala “Leonardo Torres Quevedo”, donde hemos realizado las grabaciones de sujetos sanos.

Para los sujetos patológicos, hemos utilizado la sala principal de rehabilitación del Hospital Universitario Río Hortega de Valladolid, acondicionada a las necesidades de la grabación.

Los dispositivos informáticos y electrónicos utilizados pertenecen al grupo de Telemática e Imagen de la Universidad de Valladolid y fueron adquiridos con financiación de proyectos y contratos de investigación realizados con anterioridad. Además, se emplearon dos portátiles de forma simultánea: uno para la grabación con los sensores XSens y el otro con el *software* OpenCap.

Durante las grabaciones, se han usado los sensores XSens Awinda, pertenecientes al grupo de investigación, junto con las cintas y la ropa que forma parte del kit de los propios sensores. Además, para la visión artificial, hemos empleado 2 iPads, al mismo tiempo que 2 trípodes como soporte durante la grabación.

La realización de los movimientos para el miembro superior requería una botella de agua para el movimiento “*Reach to grasp and approach*”, como son las de la Figura 16.

Así como, un libro de hojas gruesas para el movimiento de pasar dos hojas de un lado a otro, como el mostrado en la Figura 16 y un sombrero para el movimiento de “*Put on a hat/helmet*”, como el mostrado en la Figura 15.

Por último, para el movimiento de *"Open/Close a bottle and pour water"*, hemos necesitado un vaso grande y nuevamente la botella de agua de la Figura 16.



Figura 15. Libro empleado para el movimiento "Turn pages" y sombrero empleado para el movimiento de "Put on a hat/helmet"



Figura 16. Botellas y vaso empleados para el movimiento "Open/Close a bottle and pour water"

5.3 Participantes

En este estudio hemos empleado tanto sujetos controles adultos, como sujetos pediátricos patológicos, con el objetivo de medir el movimiento de los miembros superiores.

Dentro del grupo de participantes controles, fueron medidos 14 sujetos de edades entre 20–22 años, reclutados de la Escuela de Técnica Superior de Telecomunicaciones de la Universidad de Valladolid. Además, se incluyeron en la

base de datos 3 sujetos pediátricos, hermanos mellizos de algunos de los sujetos patológicos, de edades entre 7 – 11 años.

Se recopilaron datos relevantes, como la edad, el sexo, la altura, el peso, el pie y la mano dominantes y el número de pie de cada uno de los participantes, reflejado en la Tabla 1.

Apodo	Edad	Sexo	Altura	Peso	Pie dom	Mano dom	N de pie
S01	21	H	1.89	72	Derecho	Derecha	42,5
S02	21	M	1.59	66	Derecho	Derecha	38
S03	21	H	1.72	68	Derecho	Derecha	41,5
S04	21	H	1.82	84	Derecho	Derecha	43
S05	21	H	1.78	70	Derecho	Derecha	42
S06	21	H	1.80	74	Izquierdo	Derecha	43
S07	21	M	1.65	68	Derecho	Derecha	38
S08	21	M	1.74	55	Derecho	Derecha	39
S09	21	M	1.68	60	Derecho	Derecha	39
S10	21	H	1.73	80	Derecho	Derecha	43
S11	11	H	1.52	35	Derecho	Derecha	39
S12	20	M	1.60	48	Derecho	Derecha	38
S13	21	M	1.71	67	Derecho	Derecha	40
S14	22	M	1.53	50	Derecho	Derecha	37
S15	21	H	1.80	60	Derecho	Derecha	42
S16	7	M	1.27	23	Derecho	Derecha	32
S17	9	H	1.35	29	Derecho	Derecha	35

Tabla 1. Sujetos sanos participantes.

Los niños fueron seleccionados de la base de datos del Hospital Universitario Río Hortega de Valladolid, según los siguientes criterios de inclusión: (1) diagnóstico de parálisis cerebral; (2) edad de 5 a 16 años; (3) capacidad mínima para agarrar activamente un objeto. Los niños fueron excluidos en caso de cirugía previa de miembros superiores o inyecciones de toxina botulínica-A, un mes antes de la prueba.

Nuevamente, recopilamos la edad, el sexo, la altura, el peso y el número de pie de los participantes, como vemos reflejado en la Tabla 2.

Apodo	Edad	Sexo	Altura	Peso	N de pie	Patología
P01	11	H	1.52	38	39	Hemiparesia derecha
P02	6	H	1.16	18	29	Hemiparesia izquierda
P03	7	H	1.16	18	29	Hemiparesia izquierda
P04	8	H	1.29	28	33	Hemiparesia derecha
P05	8	H	1.30	31	39	Diparesia
P06	13	H	1.65	57	41	Hemiparesia izquierda
P07	5	H	1.11	20	29	Hemiparesia izquierda
P08	9	H	1.35	27	34	Hemiparesia izquierda

Tabla 2. Sujetos patológicos participantes.

5.4 Comité de Ética

Se necesitó la aprobación de dos comités éticos para grabar con los sujetos sanos adultos y con los sujetos patológicos pediátricos.

Para la captura de la base de datos de sujetos sanos se realizó en la ETSI de Telecomunicación de la Universidad de Valladolid, de acuerdo con el proyecto (PI 21-2341, Adquisición de base de datos de movimientos del cuerpo humano en actividades de la vida diaria empleando vídeo y sensores vestibles). Recibió el dictamen favorable del comité de ética de la investigación con medicamentos CEIM ÁREA DE SALUD VALLADOLID ESTE en febrero 2023.

Cada sujeto se le informó de forma oral y escrita sobre el estudio y todos ellos firmaron el consentimiento informado.

Las grabaciones con los niños diagnosticados con parálisis cerebral se realizaron en el Hospital Universitario Rio Hortega de Valladolid. Que recibió el dictamen favorable del comité de ética de la investigación con medicamentos CEIM ÁREA DE SALUD VALLADOLID OESTE en marzo 2023.

Todos los padres y niños recibieron información oral y escrita sobre el estudio antes de participar. El consentimiento informado por escrito fue firmado por los padres o tutores legales de todos los niños que fueron grabados.

La protección de la identidad de los niños capturados, al ser menores de edad, se utilizó mascarilla quirúrgica, así como gafas de sol y/o antifaces para no ser identificados.

Todos los datos personales serán tratados según lo dispuesto en el Reglamento (UE) 2016/679, de 27 de abril, General de Protección de Datos Personales y su normativa de desarrollo.

5.5 Protocolo

La adquisición de la base de datos de movimientos de miembro superior se realizará en la sala "Leonardo Torres Quevedo", mostrada en la Figura 17, perteneciente a la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad de Valladolid. En ella, se realizó la captura de movimiento de sujetos sanos adultos.



Figura 17. Sala "Leonardo Torres Quevedo"

Para la captura de datos de movimiento con sensores vestibles se emplearán los dispositivos XSens Awinda y para la visión artificial, se utilizará el *software* de código abierto OpenCap.

Se emplearán un total de 11 sensores XSens Awinda, que se colocarán en las extremidades superiores del sujeto utilizando cintas de velcro fabricadas a tal efecto. Se situarán: *head* (1), *shoulder* (2), *sternum* (1), *upper - arm* (2), *lower - arm* (2), *wrist* (2) y *pelvis* (1).

La información capturada viajará de forma inalámbrica hasta un ordenador, en la banda de 2.4 GHz. Siendo la banda de frecuencia de comunicación segura, utilizada por dispositivos *bluetooth* y *WiFi*.

Las capturas en vídeo se utilizarán 2 iPads sincronizados con un ordenador, posicionadas a 30-45° respecto a la línea en la que se ubicará el sujeto. Evitando en todo momento ubicar una de las cámaras en posición sagital con el participante de la captura.

Con cada uno de los sujetos voluntarios, se procederá de acuerdo con el siguiente protocolo:

1. Con anterioridad a la llegada del sujeto se comprueba que los sensores y los iPads tienen carga. Así mismo, se verifica la colocación del sistema de adquisición de vídeo en unas marcas destinadas a tal efecto. Además, se añadirán marcas dentro del plano de grabación para que el sujeto se encuentre centrado y una mesa centrada en él.

2. A la llegada del sujeto, se realizará una breve explicación del proyecto y se facilitará el consentimiento informado, para que lo lea con detenimiento y lo firme.
3. Cuando el sujeto firme el consentimiento, se le asignará un número de identificación y se recogerá la información relevante: edad, sexo, altura, peso, pie y mano dominante y número de pie.
4. A continuación, se procede a colocar los dispositivos vestibles junto con los sensores en las partes indicadas anteriormente. Se debe realizar tal y como se muestra en la Figura 18.



Figura 18. Colocación sensores del Sistema XSens Awinda para miembro superior

Tras la colocación, se le pide al sujeto que permanezca de pie con las piernas juntas y los brazos pegados al tronco, para realizar la calibración de los sensores. Asimismo, durante la calibración se pide al sujeto que, de unos pasos, como parte del proceso.

5. A continuación, se le pide al sujeto que se coloque en el centro del plano de grabación, en posición neutral, es decir, de pie con las piernas abiertas el ancho de las caderas y los brazos sutilmente elevados respecto al tronco, para realizar la calibración de las cámaras de los iPads.
6. Tras llevar a cabo las dos calibraciones, se le pide al sujeto que realice cada una de las actividades, especificadas en la Tabla 3. Las tres primeras actividades corresponden a análisis unimanual y las dos siguientes corresponden a un análisis bimanual. Las cinco actividades han sido seleccionadas por el interés del análisis de las articulaciones del tronco superior.

7. Una vez realizadas las capturas, se retirarán los dispositivos vestibles. Todos los archivos resultados de las grabaciones se guardarán en una carpeta que tiene nombre el identificador del sujeto.

5.6 Bloque I: Reconocimiento de actividades de la vida diaria

5.6.1 Introducción datos OpenCap

Las grabaciones de OpenCap son almacenadas en la nube personal del usuario, desde ella se puede acceder a todas las grabaciones de los sujetos y descargarse los datos de las capturas.

Cuando se descargan los datos de un sujeto, se proporciona una carpeta en formato .zip con todos los datos de interés de la grabación.

Dentro de la carpeta tenemos las imágenes de calibración de ambas cámaras empleadas durante la captura. Donde podemos observar los vectores X, Y y Z dibujados en el tablero de calibración, como se muestra en la Figura 19.

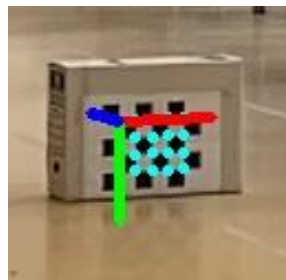


Figura 19. Tablero de calibración OpenCap con vectores X, Y y Z

Nos proporciona otra carpeta con los marcadores anatómicos y puntos clave de los vídeos, en formato .trc. Además, nos facilita otra carpeta con los datos procesados con OpenSim, tanto el modelo musculoesquelético escalado, en formato .osim, como los archivos de movimiento con los datos cinemáticos, con extensión .mot.

OpenSim es una plataforma 3D de código abierto que permite crear entornos virtuales para simular representaciones digitales de objetos y escenas del mundo real. Los archivos de movimiento se pueden cargar, junto con el modelo musculoesquelético en la interfaz de OpenSim para su visualización. También, se pueden abrir como hojas de cálculo con cualquier *software* de hojas de cálculo.

5.6.2 Movimientos de interés

La selección de movimientos que se iban a capturar se llevó a cabo apoyándonos en las escalas estándares que emplean los médicos rehabilitadores y fisioterapeutas para evaluar la funcionalidad del miembro superior.

En primer lugar, se quiso incluir varios movimientos unimanuales para evaluar los niños con hemiplejía y hemiparesia. En este caso, nos apoyamos en *The Melbourne Assessment*, una escala de valoración de la funcionalidad unimanual, que se basa en 16 movimientos. Dentro de la escala tenemos en cuenta un movimiento simple "*Reach to grasp and approach*", el paciente lo realizará primero con una mano y después con otra. Este movimiento se utilizará la abreviatura de "RTG".

Adaptamos otro de los movimientos considerados en la escala anterior un poco más complejo: "pronación-supinación". Dicho movimiento se realiza pasando varias páginas de un libro para un lado y para otro, haciéndolo más llamativo para el niño. Lo denominamos "*Turn pages*" y su abreviatura será "BOOK".

En segundo lugar, quisimos incluir actividades bimanuales más presentes en la vida diaria de los pacientes. Para ello, nos apoyamos en la *The Assisting Hand Assessment* (AHA), una escala para la valoración de tareas bimanuales en niños, que busca evaluar la funcionalidad del miembro superior afectado de forma paralela a la del miembro sano.

El movimiento desarrollado fue "*Put on a hat/helmet*", abreviado como "HAT", seleccionado por ser una actividad fácil a la hora de evaluar la simetría entre los dos lados, en pacientes con hemiplejía. Además, de ser interesante y motivadora para los niños, a su vez que desafiante a la hora de ver su capacidad.

Finalmente se seleccionó un movimiento más complejo y estimulante "*Open/Close a bottle and pour water*". Dicha actividad la podemos considerar como suma de movimientos unimanuales y bimanuales, pudiendo segmentar sus fases.

Primeramente, se realiza un movimiento unimanual para alcanzar la botella de agua, posteriormente se ejecuta un movimiento bimanual, mientras una mano abre la botella con la otra mano sujeta la botella. Después el sujeto vierte agua en el vaso, pudiendo realizarlo de manera unimanual usando una sola mano o bimanual empleando las dos. Mas adelante, se cerrará la botella considerándolo como un movimiento bimanual y finalmente se dejará la botella en la mesa con un movimiento unimanual.

Podemos observar un resumen de los movimientos en la Tabla 3, junto con la descripción detallada de las indicaciones que se darán al sujeto, el número de repeticiones del movimiento, su abreviatura y anotaciones importantes sobre los movimientos.

Id	Unilateral/Bilateral	Movimiento	Repetición	Instrucciones para el sujeto	Objetivo	Notas
UU_RTG1	Unilateral	<i>reach-to-grasp and approach RIGHT</i>	3	“Partiendo de pie con los brazos en posición natural, extienda la mano derecha para agarrar la botella y acérquela hacia el cuerpo. Después deje la botella en la mesa y vuelva a la posición inicial, con los brazos en posición natural”	Medir reaching (sensores y vídeo) y grasping (vídeo)	Se realizará con una mesa y una botella, que se colocará en frente del sujeto, cerca de la mano derecha.
UU_RTG2	Unilateral	<i>reach-to-grasp and approach LEFT</i>	3	“Partiendo de pie con los brazos en posición natural, extienda la mano izquierda para agarrar la botella y acérquela hacia el cuerpo. Después deje la botella en la mesa y vuelva a la posición inicial, con los brazos en posición natural”	Medir reaching (sensores y vídeo) y grasping (vídeo)	Se realizará con una mesa y una botella, que se colocará en frente del sujeto, cerca de la mano izquierda.
UU_BOOK	Unilateral	<i>turn pages</i>	2 lado – 2 lado	“Con el libro abierto y apoyado en la mesa, tienes que pasar una página de derecha a izquierda, dos veces y después pasar una página de izquierda a derecha dos veces”	Medir prono – supinación	Se realizará con un libro infantil con páginas rígidas de “cartón”, que se podrá apoyar en la mesa a la hora de pasar las páginas.
UB_HAT	Bilateral	<i>put on a hat/helmet</i>	3	“Con el sombrero apoyado en la mesa, tienes que agarrarlo con las dos manos y colocarlo en la cabeza. Seguidamente, tienes que separar las manos del sombrero. Después tienes que volver a coger el sombrero y dejarlo sobre la mesa”	Flexión, abducción y rotación de hombro	Se realizará con un sombrero y se partirá con el sombrero apoyado en la mesa, delante del sujeto.
UB_BOTTLE	Bilateral	<i>Open/close a bottle and pour water</i>	2	“Partiendo con los brazos relajados, tienes que coger la botella, abrirla y echar agua en el vaso. Después cierras la botella y la vuelves a colocar como en la posición inicial”	Hacer una segmentación de las fases del movimiento de echar agua	Se colocará en la mesa un vaso y una botella cerrada, colocando en línea el vaso y la botella con el sujeto, quedando más cerca del sujeto el vaso y detrás la botella.

Tabla 3. Movimientos seleccionados de miembro superior

5.6.3 Preprocesado de datos

Primeramente, antes del preprocesado de los datos de OpenCap vamos a realizar una adaptación del formato de los archivos a uno requerido por el *framework*, que vamos a emplear para entrenar las redes. Realizamos una transformación del formato de .mot a .csv.

Después, renombramos los archivos, recibiendo el nombre del sujeto al que pertenecen, el movimiento que es, considerando:

- M01: “*Reach to grasp and approach*”.
- M02: “*Turn pages*”.
- M03: “*Put on a hat*”.
- M04: “*Open/Close a bottle and pour water*”.

Posteriormente, con el objetivo de conseguir más datos, dividimos cada archivo .csv en periodos ventana de 60 *frames* con un solapamiento de 30. Al ser movimientos relativamente largos emplearemos tamaños de ventanas de 1 segundo.

Dentro de estos archivos hacemos una selección de los ángulos que nos interesan para el reconocimiento de patrones en el miembro superior. Consideramos todos los ángulos relacionados con los brazos: “*arm_flex_r*”, “*arm_add_r*”, “*arm_rot_r*”, “*elbow_flex_r*”, “*pro_sup_r*”, “*arm_flex_l*”, “*arm_add_l*”, “*arm_rot_l*”, “*elbow_flex_l*”, “*pro_sup_l*”, un total de 10 ángulos.

Dentro de los ángulos que han sido descartados, debido a su baja influencia en la realización del movimiento, tenemos: “*lumbar_extension*”, “*lumbar_bending*”, “*lumbar_rotation*”, “*pelvis_tilt*”, “*pelvis_list*”, “*pelvis_rotation*”, “*hip_flexion_r*”, “*hip_adduction_r*”, “*hip_rotation_r*”, “*knee_angle_r*”, “*ankle_angle_r*”, “*subtalar_angle_r*”, “*mtp_angle_r*”, “*hip_flexion_l*”, “*hip_adduction_l*”, “*hip_rotation_l*”, “*knee_angle_l*”, “*ankle_angle_l*”, “*subtalar_angle_l*”, “*mtp_angle_l*”, “*lumbar_extension*”, “*lumbar_bending*”, “*lumbar_rotation*”, “*lumbar_extension*”, “*lumbar_bending*”, “*lumbar_rotation*”.

5.7 Bloque II: Segmentación de las fases de “*Open/Close a bottle and pour water*”.

5.7.1 Introducción y fases de *Open/Close a bottle and pour water*.

Hemos adquirido una base de datos de movimiento del tren superior bastante amplia. Sin embargo, nos centraremos en el movimiento de *Open/Close a bottle and pour water*. Este movimiento es de gran interés, al deberse de un movimiento complejo, compuesto por diversas fases, ya sean unimanuales y/o bimanuales. Además, es una actividad que puede estar presente en la vida diaria de los pacientes con parálisis cerebral.

Nuestro objetivo con el movimiento de *Open/Close a bottle and pour water* es segmentar las fases presentes durante la realización de este complejo movimiento.

Posteriormente, las etiquetaremos en cada vídeo capturado para, después, entrenar una red neuronal con dicho conjunto de datos, y que sea capaz de identificar que fase está realizando el sujeto. Aunque este pueda ser un sujeto sano o pediátrico patológico.

El movimiento comienza con un vaso y una botella sobre una mesa en frente del sujeto, colocados en orden de cercanía a él, primero el vaso, y detrás, la botella de agua.

El participante comenzará la actividad cogiendo la botella de agua, abriéndola, y después echando agua en el vaso. Para más adelante, cerrarla y volver a situar la botella y el vaso en su posición inicial, tal y como se ha mencionado, primero el vaso y detrás, la botella de agua.

Hemos considerado un total de 5 fases durante el desarrollo de este movimiento, que se muestran en la Tabla 4. Sin embargo, en sujetos pediátricos, ya fueran sanos o patológicos, se incluyó una sexta etiqueta, correspondiente al movimiento de echar agua sujetando la botella con las dos manos. Dicha etiqueta fue denominada, "2Manos".

Una vez definidas las diferentes fases dentro del movimiento, se establecieron unos criterios para etiquetar de forma uniforme los vídeos, es decir, se definieron los límites de inicio y finalización de cada submovimiento, tal y como se muestra en la Tabla 4.

El etiquetado de los movimientos anteriormente mencionados se realizará asignando el número que se le ha fijado a cada submovimiento.

Nº	Nombre movimiento	Descripción	Criterio de inicio	Criterio de finalización	Notas
0	<i>No movimiento</i>	Cuando el sujeto no realiza ninguna de las fases de interés	Se establece por defecto cuando no se etiqueta ninguna de las fases de interés		Se define por defecto cuando no se establece ninguna de las otras fases
1	<i>Acercar/Alejar Zoom in/out</i>	El sujeto acerca la mano a la botella, la agarra y la acerca al cuerpo. De forma análoga, tras echar agua, aleja la botella del cuerpo y la deja sobre la mesa	Comienza cuando el brazo del sujeto sobrepasa la altura de su cadera	Finaliza cuando la mano que ha dejado la botella vuelve a una posición relajada, paralela al tronco	Hemos considerado Acercar/Alejar el mismo movimiento, los dos movimientos son muy similares, pero en distinta dirección
2	<i>Transferir Transfer</i>	El sujeto cambia de mano la botella para después abrir/cerrar la botella o dejarla sobre la mesa	Comienza cuando la botella se encuentra vertical y la mano que recibe la botella se encuentra preparada	Finaliza cuando la botella está en la mano que recibe la botella y esta se encuentra en posición vertical	Pueden no realizar este movimiento, realizarlo o incluso realizarlo dos veces, durante la grabación
3	<i>Sujetar Hold</i>	Se realiza de forma simultánea al movimiento de abrir/cerrar.	Comienza a la vez que el movimiento Abrir/Cerrar	Finaliza cuando el sujeto separa por complejo la mano de la botella	Se considera sujetar la botella al abrir/cerrar y al vaso cuando se echa agua
4	<i>Abrir/Cerrar Open/Close</i>	El sujeto abre la botella o la cierra para echar agua al vaso	Comienza cuando el sujeto posiciona la mano en el tapón de la botella	Finaliza cuando el sujeto separa por completo la mano del tapón de la botella	Hemos considerado Abrir/Cerrar el mismo movimiento, los dos movimientos son muy similares, pero en distinta dirección
5	<i>Verter Pour</i>	El sujeto inclina la botella y echa agua al vaso	Comienza cuando el sujeto inclina la botella	Finaliza cuando el sujeto comienza a poner vertical la botella	

Tabla 4. Fases y límites propuestos para la segmentación del movimiento Open/Close a bottle and pour water

5.7.2 Etiquetado de datos

El proceso de etiquetado se llevará a cabo mediante una interfaz desarrollada en Visual Studio Code, empleando el lenguaje de programación Python. La creación de una interfaz nos facilitará tanto el etiquetado de los vídeos como su visualización, desde la perspectiva de las dos cámaras, pudiendo visualizar ambos vídeos de forma simultánea.

La interfaz se compone de 6 *frames*, como mostramos en la Figura 20: cuatro de ellos destinados para la visualización y reproducción de los vídeos, uno para colocar los botones de reproducción y el último para colocar las etiquetas del movimiento *Open/Close a bottle and pour water*.

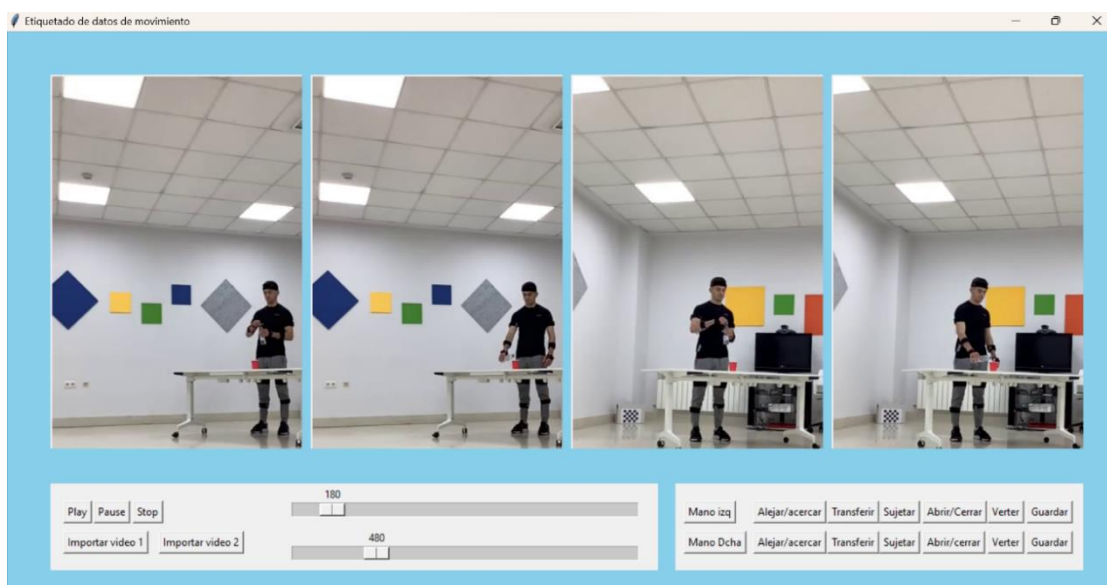


Figura 20. Interfaz desarrollada para el etiquetado de datos

Dentro del *frame* de botones de reproducción, cuenta con los botones de "Play", "Pause", "Stop", "Importar vídeo 1", "Importar vídeo 2" y 2 *sliders* para avanzar los vídeos *frame a frame*, podemos apreciar el diseño en la Figura 21.

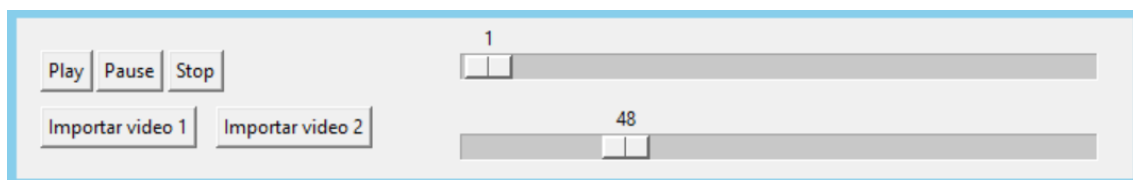


Figura 21. Frame de la interfaz con los botones de reproducción de los vídeos.

En el *frame* de etiquetas se ha creado tanto para la extremidad derecha como izquierda las etiquetas de los submovimientos: "Acercar/Alejar", "Transferencia", "Sujetar", "Abrir/Cerrar" y "Verter". Además, se incluyen dos botones para guardar las etiquetas para cada extremidad, como podemos observar en la Figura 22.

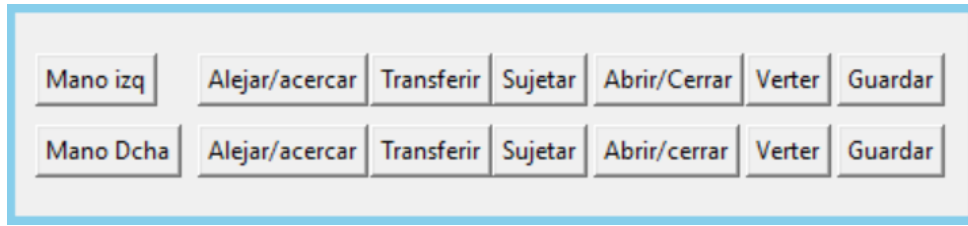


Figura 22. Frame de la interfaz con los botones de etiquetado.

Se reproduce un vídeo desde las dos cámaras para facilitar su visualización desde diferentes perspectivas. Con ambos *sliders* podemos desplazar el vídeo de forma pareada: con el slider de arriba se desplaza el vídeo en el primer y tercer *frame*, desde las cámaras 1 y 2 y con el slider de abajo se desplaza el vídeo en el segundo y cuarto desde ambas cámaras.

Su funcionamiento se basa en desplazar el vídeo con los *sliders*, una vez que situemos el inicio del submovimiento en el primer slider y el final en el segundo slider, seleccionamos la etiqueta que corresponde y le damos a guardar a la extremidad que pertenezca. Este proceso lo repetiremos etiquetando cada movimiento que aparezca en el vídeo.

Cuando terminemos el etiquetado, al cerrar el programa se generará un documento de texto para cada extremidad con las etiquetas seleccionadas.

Tras las grabaciones con los sujetos patológicos hemos añadido una etiqueta extra, al realizar el movimiento de verter agua sujetando la botella con las dos manos. Asignándole el número 6 y denominándola “2Manos”, como vemos incluida en la Figura 23.

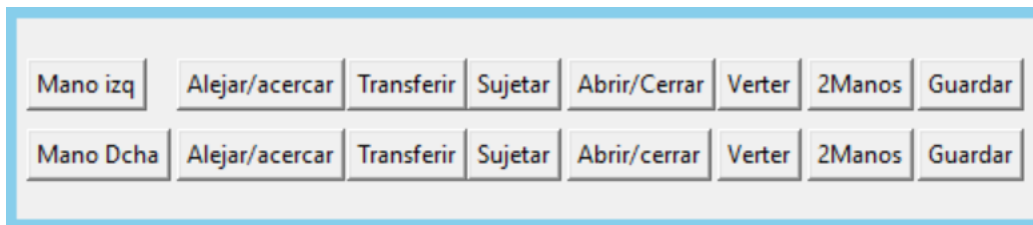


Figura 23. Frame de la interfaz para etiquetar a los sujetos patológicos

Una vez que finalizamos de etiquetar el vídeo al salir de la interfaz, nos genera dos documentos de texto: uno para la extremidad superior derecha y otro para la extremidad izquierda.

Una vez etiquetados los vídeos, unimos los archivos de las etiquetas con los archivos “mot” que proporciona OpenCap tras la grabación. Los archivos “mot” contienen los datos cinemáticos de los movimientos capturados. En concreto, los ángulos de varias articulaciones, en grados.

5.7.3 Preprocesado de datos

En esta fase se separan las muestras de entrenamiento en base al etiquetado: etiquetas del lado derecho (R), denominadas “EtiquetaDer” y etiquetas del lado izquierdo (L), llamadas “EtiquetaIzq”.

Una vez separadas, se volverán a dividir por etiquetas de movimientos, creando un archivo para cada uno de ellos en formato .csv y renombrándolos de forma específica. Cada uno recibirá el nombre del sujeto al que pertenecen, si pertenecen al lado derecho (R) o al lado izquierdo (L), y la etiqueta del movimiento correspondiente.

Posteriormente, con el objetivo de conseguir más datos, dividimos cada archivo .csv en periodos ventana de 30 muestras o *frames* con un solapamiento de 15 muestras. Consideramos dicho tamaño al ser el tiempo correspondiente a medio segundo.

Dentro de estos archivos hacemos una selección de los ángulos que nos interesan para el reconocimiento de patrones en el miembro superior. Consideramos todos los ángulos relacionados con los brazos: "arm_flex_r", "arm_add_r", "arm_rot_r", "elbow_flex_r", "pro_sup_r", "arm_flex_l", "arm_add_l", "arm_rot_l", "elbow_flex_l", "pro_sup_l", un total de 10 y los 3 correspondientes a la espalda: "lumbar_extension", "lumbar_bending", "lumbar_rotation". Debido a cierta inclinación de los sujetos a la hora de coger la botella y el vaso, hemos tenido en cuenta los tres movimientos de la parte baja de la espalda.

Dentro de los ángulos que han sido descartados, debido a su baja influencia en la realización del movimiento, tenemos: "lumbar_extension", "lumbar_bending", "lumbar_rotation", "pelvis_tilt", "pelvis_list", "pelvis_rotation", "hip_flexion_r", "hip_adduction_r", "hip_rotation_r", "knee_angle_r", "ankle_angle_r", "subtalar_angle_r", "mtp_angle_r", "hip_flexion_l", "hip_adduction_l", "hip_rotation_l", "knee_angle_l", "ankle_angle_l", "subtalar_angle_l", "mtp_angle_l", "lumbar_extension", "lumbar_bending", "lumbar_rotation".

6. RESULTADOS Y DISCUSIÓN

6.1 Resultados

6.1.1 Bloque I: Reconocimiento de actividades de la vida diaria

Los datos capturados con OpenCap ya tratados están divididos en los movimientos seleccionados anteriormente, “*Reach to grasp a bottle and approach*”, “*Turn pages*”, “*Put on a hat*” y “*Open/Close a bottle and pour water*”. Usando los ángulos mencionados previamente, correspondientes únicamente a los brazos.

Además, tomamos la decisión de juntar los movimientos de “*Reach to grasp a bottle and approach*”, ya fueran del lado derecho o del lado izquierdo, unificándolos.

Emplearemos las abreviaturas de las actividades durante los entrenamientos para simplificar sus nombres. Por lo que, “*Reach to grasp a bottle and approach*” será “RTG”, “*Turn pages*” como “BOOK”, “*Put on a hat/helmet*” como “HAT” y “*Open/Close a bottle and pour water*” como “BOTTLE”.

Anteriormente en la etapa de preprocesado, hemos dividido los datos en ventanas de 60 con un solapamiento de 30, es decir, en ventanas de un segundo. En primer lugar, definimos una configuración inicial de la red neuronal, que será común para los entrenamientos con sujetos sanos, pediátricos y todos juntos.

En base a que no se dispone de muchas muestras de entrenamiento y algún experimento previo, se eligió un tamaño de *batch* de 4 ventanas.

Usamos la técnica de *Early Stopping* para evitar el *overfitting* durante el proceso de entrenamiento. A medida que se entrena la red neuronal, se comprueba el rendimiento del *loss* de validación, en caso de que empore durante 10 épocas, detiene el entrenamiento evitando el sobre entrenamiento y reduciendo el tiempo de entrenamiento.

Establecimos un número de 30 épocas. Sin embargo, en caso de que no se finalizara el entrenamiento previamente debido a *Early Stopping* (llegando a las 30 épocas) se aumentarían hasta 50, para ver la evolución del modelo en su totalidad.

También, empleamos el método de *Reduce on Plateau*, que reduce la tasa de aprendizaje una vez que se estanca la métrica de interés, en este caso, la función de pérdidas del set de validación.

Además, utilizamos el optimizador “*Adam*”, muy empleado para problemas de este tipo, y la función de pérdida “*Categorical cross entropy*”, habitual en el entrenamiento de redes. La métrica que se va a evaluar durante todo el proceso será la precisión (*accuracy*).

6.1.1.1 ENTRENAMIENTO CON SUJETOS CONTROLES

En primer lugar, comenzamos con una red convolucional de 4 capas convolucionales de 32, 64, 128 y 256 filtros y 5 capas de *MaxPooling*. Compuesta, también, por una capa Dense de 256 neuronas. El modelo final cuenta con un total de 1.215.976 parámetros, como podemos observar en la Figura 24.

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)             (None, 60, 10, 32)         320
max_pooling2d (MaxPooling2D) (None, 30, 5, 32)          0
batch_normalization (BatchN (None, 30, 5, 32)          120
ormalization)
conv2d_1 (Conv2D)           (None, 30, 5, 64)          32832
max_pooling2d_1 (MaxPooling (None, 15, 3, 64)          0
2D)
batch_normalization_1 (Batc (None, 15, 3, 64)          60
hNormalization)
conv2d_2 (Conv2D)           (None, 15, 3, 128)         131200
max_pooling2d_2 (MaxPooling (None, 8, 2, 128)          0
2D)
batch_normalization_2 (Batc (None, 8, 2, 128)          32
hNormalization)
conv2d_3 (Conv2D)           (None, 8, 2, 256)          524544
max_pooling2d_3 (MaxPooling (None, 4, 1, 256)          0
2D)
batch_normalization_3 (Batc (None, 4, 1, 256)          16
hNormalization)
dropout (Dropout)           (None, 4, 1, 256)          0
flatten (Flatten)           (None, 1024)                0
dense (Dense)                (None, 512)                 524800
dense_1 (Dense)              (None, 4)                   2052
-----
Total params: 1,215,976
Trainable params: 1,215,862
Non-trainable params: 114

```

Figura 24. Red neuronal convolucional de 4 capas de 32, 64, 128 y 256 filtros.

El entrenamiento se para en la época 19. Los resultados del set de *train*, tenemos un *loss* de 0.011 y una *accuracy* de 0.996. Mientras, en el set de validación un *loss* de 0.643 y una *accuracy* de 0.843, podemos ver que la red tiene un rendimiento muy bueno con el set de validación. En el set de *test*, tenemos un *loss* de 0.722 y una *accuracy* de 0.875. Se obtuvieron las métricas de rendimiento mostradas en la Tabla 5.

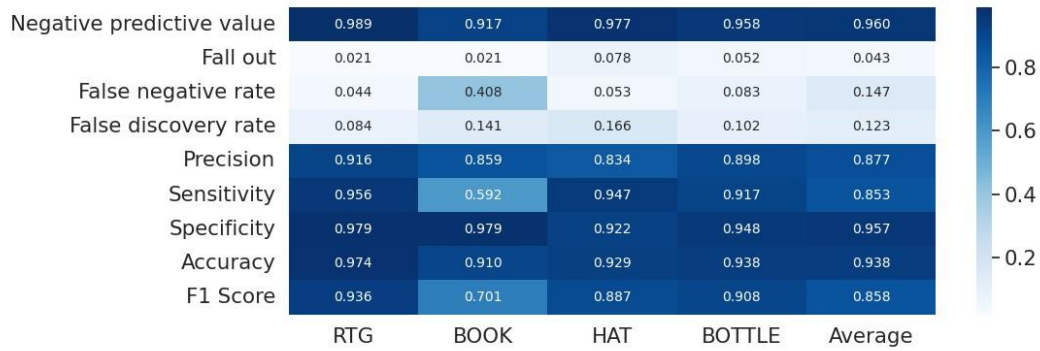


Tabla 5. Métricas de rendimiento del entrenamiento de sujetos controles para todos los movimientos

Podemos observar buenos resultados en la gran mayoría de las etiquetas, con una sensibilidad y especificidad alta. Sin embargo, señalamos que la sensibilidad de la etiqueta “BOOK”, es relativamente baja. Para entenderlo mejor nos fijamos en la matriz de confusión de la Figura 25.

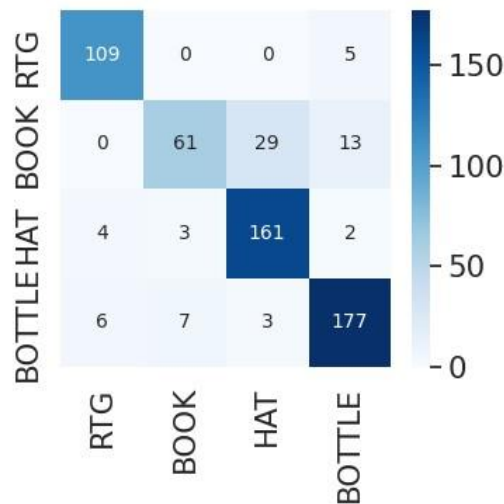


Figura 25. Matriz de confusión de la Tabla 5

La etiqueta “BOOK”, presenta una sensibilidad baja y una especificidad alta. Por lo que tienen una baja capacidad para detectar verdaderos positivos y una alta capacidad para reconocer los casos negativos y evitar falsos positivos.

Además, tiene una precisión alta, por lo que confirma que la red tiene una tasa baja de falsos positivos y clasifica correctamente la mayoría de los casos positivos.

Respaldo esta idea, la matriz de confusión de la Figura 25 muestra 71 casos clasificados como “BOOK”, cuando han sido bien clasificados 61. Señalamos que la red confunde la actividad de “BOOK” con otras etiquetas como “HAT” y “BOTTLE”. Esto puede deberse a la primera parte de los movimientos genere confusión, al comenzar los movimientos de manera similar.

No obstante, el rendimiento de la red sigue siendo bueno con el resto de las etiquetas, obteniendo muy buenos resultados con el set de *test*.

6.1.1.2 ENTRENAMIENTO CON SUJETOS PEDIÁTRICOS.

Comenzamos el entrenamiento utilizando una red convolucional de 4 capas convolucionales de 32, 64, 128 y 256 filtros y 5 capas de *MaxPooling* intercaladas.

Esta vez realizaremos un *K-Fold* con los 9 sujetos que disponemos. Este proceso se repetirá durante 5 veces, variando la presencia de los sujetos en los diferentes grupos: conjunto de *train*, validación y *test*.

El objetivo es obtener una estimación más precisa y robusta del rendimiento del modelo, al tener en cuenta la variabilidad de los datos de *train* y *test*. Realizamos este proceso al disponer muy pocos sujetos pediátricos y no disponer de más datos. Con dicho método evitamos el sobre ajuste y proporcionamos una evaluación más fiable del modelo, al poder evaluar subconjuntos de datos.

En la Tabla 6 vemos reflejadas las métricas de rendimiento obtenidas. En los movimientos, "RTG" y "HAT", observamos una sensibilidad y una especificidad relativamente alta con una precisión muy parecida.

Negative predictive value	0.934	0.881	0.934	0.689	0.860
Fall out	0.086	0.161	0.082	0.229	0.139
False negative rate	0.237	0.637	0.296	0.432	0.401
False discovery rate	0.212	0.691	0.291	0.323	0.379
Precision	0.788	0.309	0.709	0.677	0.621
Sensitivity	0.763	0.363	0.704	0.568	0.599
Specificity	0.914	0.839	0.918	0.771	0.861
Accuracy	0.878	0.766	0.877	0.682	0.801
F1 Score	0.734	0.306	0.671	0.613	0.581
	RTG	BOOK	HAT	BOTTLE	Average

Tabla 6. Métricas de rendimientos pediátricos con todos los movimientos.

Nos apoyamos en la matriz de confusión de la Figura 26, vemos como la red ha predicho 714 ventanas como "RTG", cuando verdaderamente lo eran 510. Mientras que para la etiqueta "HAT", se clasifica un total de 592, siendo realmente 387. Por lo que indica que la red está generando tanto falsos positivos como falsos negativos.

Para otras etiquetas como "BOTTLE", ocurre un proceso similar, pero con una precisión sutilmente más baja. Si nos fijamos en la matriz de confusión de la Figura 26, se clasifica como la etiqueta de "BOTTLE" un total de 1122, de las cuales verdaderamente son 745, con un total de 578 falsos negativos.

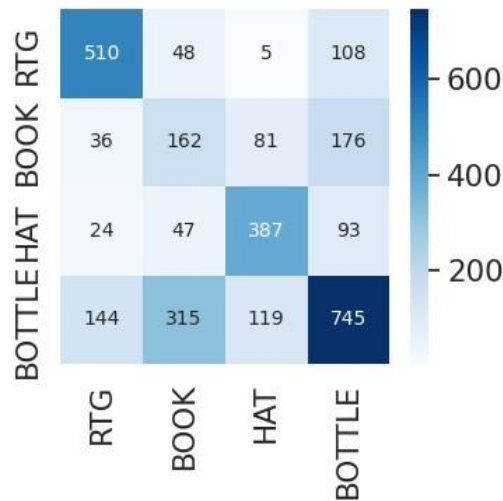


Figura 26. Matriz de confusión de la Tabla 6.

Observamos que la etiqueta con peores métricas es “BOOK”, con una sensibilidad baja y una especificidad alta sumado a una precisión baja. Esto lo vemos reflejado en la matriz de confusión de la Figura 26, la red confunde la etiqueta de “BOOK” con la etiqueta de “BOTTLE”, generando un elevado número de falsos positivos.

Dentro de los *K-Fold* realizados seleccionamos el conjunto que ha obtenido mejores resultados y lo entrenamos con una red con una capa *Dense* final, aumentando la cantidad de parámetros entrenables.

Obtenemos en el set de *train*, un *loss* de 0.075 y una *accuracy* 0.972 y en el set de validación, un *loss* de 1.799 y una *accuracy* de 0.680. Si nos fijamos en los resultados del set de *test* tenemos un *loss* de 1.716 y una *accuracy* de 0.788.

Las métricas de rendimiento reflejadas en la Tabla 7, en comparación con el entrenamiento anterior, han mejorado. Nuevamente, examinamos las etiquetas de “RTG” y “HAT”, señalando que la sensibilidad ha aumentado.

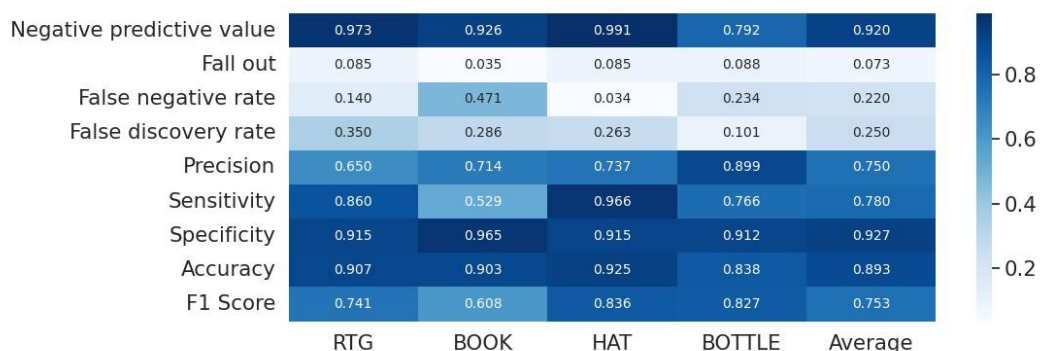


Tabla 7. Métricas de rendimiento del *K-Fold* realizado a los sujetos pediátricos.

Para confirmarlo, nos fijamos en la matriz de confusión de la Figura 27, la etiqueta “RTG” ha sido clasificado 123 veces, de las cuales 80 sí lo eran. Además, la tasa de falsos negativos ha sido muy baja, equivocándose en 13 ventanas, solamente.



Figura 27. Matriz de confusión de la Tabla 7.

Para la etiqueta “HAT”, 156 ventanas han sido categorizadas como tal, de las cuales 115 realmente lo eran. Observamos que ha confundido etiquetas de “BOOK” como “HAT”.

De nuevo, la etiqueta de “BOOK” presenta las peores métricas, sin embargo, con respecto al entrenamiento anterior, ha mejorado su precisión. La red ha categorizado 63 ventanas como dicha etiqueta, siendo verdaderas 45. Vemos que han disminuido los falsos positivos, pero la tasa de falsos negativos sigue siendo alta, confundiendo “BOOK” con “HAT”.

Esto puede deberse a que el movimiento que más dificultades presentaba para los pacientes era “BOOK”, al ser una actividad que suponía realizar pinza con los dedos para pasar página y un movimiento de pronosupinación con el brazo.

Por último, “BOTTLE” presenta buenas métricas, aunque tiene una sensibilidad discretamente baja. Esto se ve reflejado en una tasa de falsos negativos sutilmente elevada, un total de 71 ventanas clasificándolas como otras etiquetas.

6.1.1.3 ENTRENAMIENTO SUJETOS CONTROLES Y PEDIÁTRICOS: SANOS Y PATOLÓGICOS.

Procedemos a utilizar la red anterior, una red convolucional de 4 capas de 32, 64, 128 y 256 filtros y 5 capas de *MaxPooling*. Compuesta, también, por una capa *Dense* de 256 neuronas. Dotando a la red de un total de 1.215.976 parámetros entrenables.

Tras el entrenamiento conseguimos un *loss* de 0.203 y una *accuracy* de 0.993 en el set de *train*, mientras que en el set de validación obtenemos un *loss* de 0.934 y una *accuracy* de 0.826. Si observamos el set de *test* obtenemos un *loss* de 2.002 y una *accuracy* de 0.750.

Procedemos a analizar las métricas de rendimiento, reflejadas en la Tabla 8, conseguidas en dicho entrenamiento. Las etiquetas de “RTG” y “HAT” presentan sensibilidad, especificidad y precisión muy similares, siendo buenas.

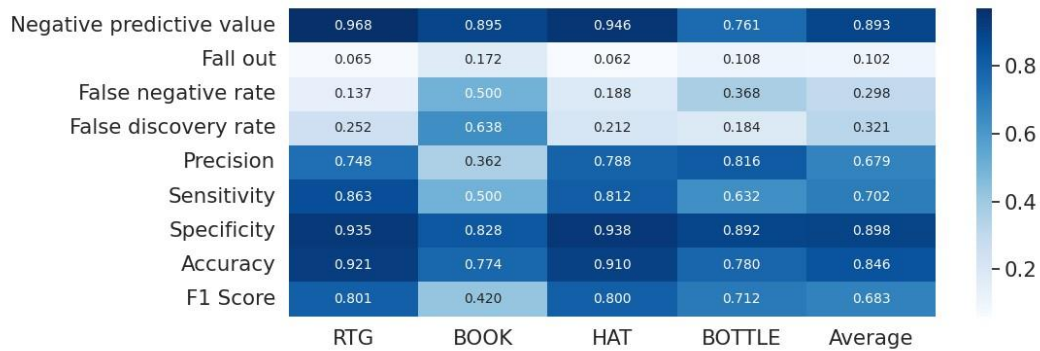


Tabla 8. Métricas de rendimiento de entrenamiento sujetos controles y patológicos.

Podemos verificarlo, fijándonos en la matriz de confusión de la Figura 28. La etiqueta de “RTG” ha sido clasificada 286 veces, de las cuales fueron bien clasificadas 214 de ellas y para “HAT” fueron categorizadas como tal, un total de 307, de las cuales 242 eran verdaderas. Ambos, presentan una tasa de falsos negativos bastante baja.



Figura 28. Matriz de confusión de la Tabla 8.

Para otras etiquetas como “BOOK”, la sensibilidad obtenida es baja como en otros entrenamientos, mientras que la especificidad es alta, junto con una precisión baja. Esto indica que la red no la clasifica bien y la confunde con otras etiquetas. Nos apoyamos en la matriz de confusión para comprobarlo.

Vemos como “BOOK” ha sido clasificado un total de 304, de las cuales, sí eran 110, generando 65 falsos positivos y un total de 110 falsos negativos.

En resumen, volvemos a ver que la etiqueta “BOOK” es la que más problemas da ser reconocida por la red. Esto se debe a las dificultades que presentaron los niños para realizar dicha actividad.

Por último, la etiqueta “BOTTLE” presenta una especificidad alta y una sensibilidad baja, así como una precisión baja. Lo vemos reflejado en la matriz de confusión de la Figura 28, la red ha clasificado 451 ventanas como tal, siendo realmente la verdaderas 368.

Podemos observar los resultados obtenidos en los experimentos de Bloque I: “reconocimiento de actividades de la vida diaria”, en la Tabla 9.

RESULTADOS BLOQUE I: RECONOCIMIENTO DE ACTIVIDADES DE LA VIDA DIARIA

Tipo de sujeto	Grupo al que pertenecen			Red neuronal				Resultados obtenidos			
	Set <i>train</i>	Set <i>valid.</i>	Set <i>test</i>	Tipo de red	Nº de capas	Parámetros entrenables	Técnicas de regularización	Parada (<i>epoch</i>)	Set de <i>train</i>	Set de <i>valid.</i>	Set de <i>test</i>
CONTROLES	"S01", "S02", "S03", "S04", "S05", "S06", "S07", "S08", "S09", "S12"	"S13" y "S14"	"S10" y "S15"	CNN.	4 CNN.: 32, 64, 128 filtros + Dense: 256	1.215.976	DropOut + Batch Normalization	19	Loss: 0.011 Accuracy: 0.996	Loss: 0.643 Accuracy: 0.843	Loss: 0.722 Accuracy: 0.875
PEDIÁTRICOS	"P01", "P02", "P03", "P04", "P05"	"S17"	"S16" y "P08"	CNN.	4 CNN: 32, 64, 128 filtros + Dense: 256	1.215.976	DropOut + Batch Normalization	15	Loss: 0.075 Accuracy: 0.972	Loss: 1.799 Accuracy: 0.680	Loss: 1.716 Accuracy: 0.788
CONTROLES + PEDIÁTRICOS	"S01", "S02", "S03", "S04", "S05", "S06", "S07", "S08", "S09", "S12", "P01", "P02", "P03", "P04"	"S13" "S14" y "S17"	"S10", "S15", "S16", "P05" y "P08"	CNN.	4 CNN: 32, 64, 128 filtros + Dense: 256	1.215.976	DropOut + Batch Normalization	14	Loss: 0.025 Accuracy: 0.991	Loss: 1.544 Accuracy: 0.742	Loss: 2.169 Accuracy: 0.680

Tabla 9. Resumen de los resultados de los experimentos Bloque I: Reconocimiento de actividades de la vida diaria.

6.1.2 Bloque II: Segmentación de las fases del movimiento Open/Close a bottle and pour water.

Los datos capturados con OpenCap ya tratados están divididos en los movimientos seleccionados anteriormente, para el lado derecho y para el lado izquierdo. Usando los ángulos mencionados previamente, correspondientes a los brazos y a la parte baja de la espalda, las lumbares.

Tomamos la decisión de entrenar una red neuronal para cada lado del cuerpo, con el objetivo de que cada una sea capaz de reconocer los movimientos propuestos para cada extremidad y no le genere confusión la realización simultánea de los movimientos durante la actividad de “Open/Close a bottle and pour water”.

Anteriormente en la etapa de preprocesado, en la que dividimos los datos en ventanas de 30 *frames* con un solapamiento de 15, nos dimos cuenta de que el movimiento “transferir”, correspondiente a la etiqueta número 2, era una actividad muy corta y con una baja representatividad entre los sujetos. Por lo que fue descartada y eliminada de la red neuronal, con la finalidad de que esta no interfiriera en los resultados.

En primer lugar, establecemos una configuración inicial a la red neuronal que, en caso de que no se indique lo contrario, será común en la mayoría de los entrenamientos.

Para los entrenamientos elegimos un *batch* de 4, ya que, al ser un tamaño pequeño ayudamos a la red a generalizar mejor.

Usamos la técnica de *Early Stopping* para evitar el *overfitting* durante el proceso de entrenamiento. A medida que se entrena la red neuronal, se comprueba el rendimiento del *loss* de validación, en caso de que empore durante 10 épocas, detiene el entrenamiento evitando el sobre entrenamiento y reduciendo el tiempo de entrenamiento. También, empleamos el método de *Reduce on Plateau*, que reduce la tasa de aprendizaje una vez que se estanca la métrica de interés, en este caso, la función de pérdidas del set de validación.

Establecimos un número de 30 épocas. Sin embargo, en caso de que no se finalice el entrenamiento debido al uso de *Early Stopping* (llegando a las 30 épocas), se aumenta el número de épocas hasta 50, para ver la evolución del modelo en su totalidad.

Además, utilizamos el optimizador “Adam”, muy empleado para problemas de este tipo, y la función de pérdida “*categorical cross entropy*”, habitual en el entrenamiento de redes. La métrica que se va a evaluar durante todo el proceso será la precisión (*accuracy*).

6.1.2.1 ENTRENAMIENTO CON REDES DENSE CON SUJETOS CONTROLES.

Experimento 1: selección del tamaño de ventana y solapamiento.

I. Experimento 1.1: utilización del tamaño de ventana 30.

El entrenamiento lo comenzamos probando una red neuronal sencilla, un modelo secuencial con 4 capas *Dense*, con 32, 64, 128 y 32 capas de entrada, con la función de activación "*relu*". Dotando al modelo con 27.204 parámetros entrenables, como podemos observar en el esquema de la Figura 29 de la red neuronal.

```
Model: "sequential"
-----
Layer (type)                Output Shape              Param #
-----
flatten (Flatten)           (None, 390)               0
dense (Dense)                (None, 32)               12512
dense_1 (Dense)              (None, 64)               2112
dense_2 (Dense)              (None, 128)              8320
dense_3 (Dense)              (None, 32)               4128
dense_4 (Dense)              (None, 4)                132
-----
Total params: 27,204
Trainable params: 27,204
Non-trainable params: 0
-----
```

Figura 29. Red neuronal con 4 capas *Dense* de 32, 64, 128 y 32 neuronas.

Finaliza el entrenamiento con un *loss* de 0.008 y una *accuracy* de 0.997 en el set de *train* y un *loss* de 0.310 y una *accuracy* de 0.857 del set de validación. Esto indica la presencia de *overfitting*, donde el modelo se ha ajustado demasiado a los datos de entrenamiento y no es capaz de generalizar bien con los datos nuevos. Sin embargo, también destacamos el escaso número de parámetros entrenables de la red, lo que señala que su bajo rendimiento se puede deber a esto.

Para confirmar si se produce *overfitting*, empleamos el conjunto de *test*, obtenemos un *loss* de 4.390, siendo considerablemente alto y una precisión de 0.450, baja en comparación con los resultados del conjunto de *train* y validación. Estos resultados indican que el modelo no generaliza bien y respaldan la presencia de *overfitting*.

Realizamos un entrenamiento con los movimientos del lado izquierdo, reutilizando la red neuronal anteriormente comentada, representada en la Figura 29. Observamos que el modelo obtiene un *loss* de 0.004 y una *accuracy* de 1.000 en el set de entrenamiento y en el set de validación un *loss* 1.714 y una *accuracy* de 0.714. Por lo tanto, nuevamente se produce *overfitting* con los datos del lado izquierdo.

No obstante, tampoco descartamos la idea de aumentar el número de parámetros de la red con el objetivo de mejorar el entrenamiento y solventar los problemas. Por ello, modificamos la red neuronal aumentando los parámetros de entrada de la última capa, pasando de 32 a 256, consiguiendo que aumenten hasta 56.996 parámetros entrenables, mostrada dicha estructura en la Figura 30. Volvemos a entrenar la red neuronal empleando los movimientos del brazo derecho.

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
flatten (Flatten)           (None, 390)                 0
dense (Dense)                (None, 32)                 12512
dense_1 (Dense)             (None, 64)                 2112
dense_2 (Dense)             (None, 128)               8320
dense_3 (Dense)             (None, 256)              33024
dense_4 (Dense)             (None, 4)                 1028
-----
Total params: 56,996
Trainable params: 56,996
Non-trainable params: 0
-----

```

Figura 30. Red neuronal 4 capas Dense de 32, 64, 128 y 256 neuronas.

Sin embargo, el entrenamiento se para en la época 19 debido a la técnica *Early Stopping*. Los resultados son muy similares a la anterior, obtenemos un *loss* de $6.64e-4$ y una *accuracy* de 1 en el set de *train*, y un *loss* de 0.894 y una *accuracy* de 0.805 en el set de validación.

Si nos fijamos en el set de test, hemos conseguido un *loss* de 2.056 y una *accuracy* de 0.727. Lo que nos indica que la red sigue sobre entrenando, ajustándose a los datos de entrada, sin generalizar a datos nunca vistos.

Fijándonos en la Tabla 10, el movimiento de abrir/cerrar cuenta con una sensibilidad alta y una especificidad baja, lo que indica que es capaz de detectar los casos positivos, asimismo, tiene dificultades para identificar los casos negativos, siendo la red propensa a generar falsos positivos. La precisión de esta etiqueta es baja lo que nos vuelve a señalar que los resultados clasificados como positivos son falsos positivos.

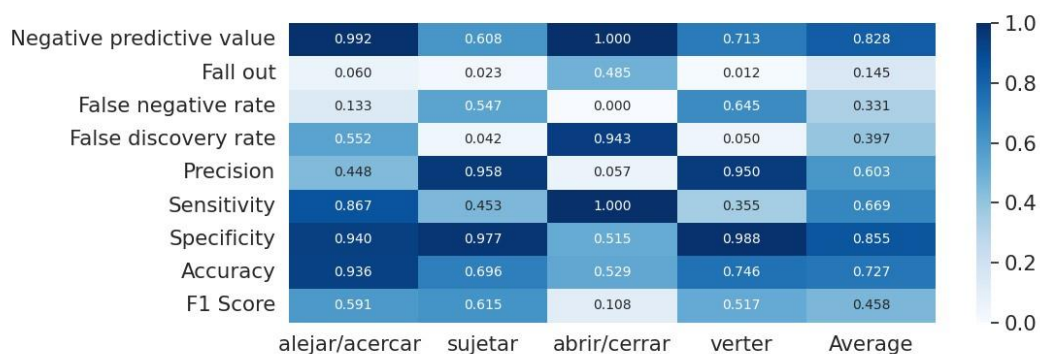


Tabla 10. Métricas de rendimiento con ventanas de 30 frames.

Para movimientos como sujetar y verter, presenta una sensibilidad baja y una especificidad alta. Esto significa que ambas tienen una baja capacidad para detectar verdaderos positivos y una alta capacidad para reconocer los casos negativos y evitar falsos positivos.

Además, tiene una precisión alta, por lo que confirma que la red tiene una tasa baja de falsos positivos y clasifica correctamente la mayoría de los casos positivos. Respalda esta idea, la matriz de confusión de la Figura 31 muestra 71 casos clasificados como sujetar, cuando han sido bien clasificados 68 y con la etiqueta de verter 40 han sido categorizados como tal, cuando en realidad 38 sí lo son.

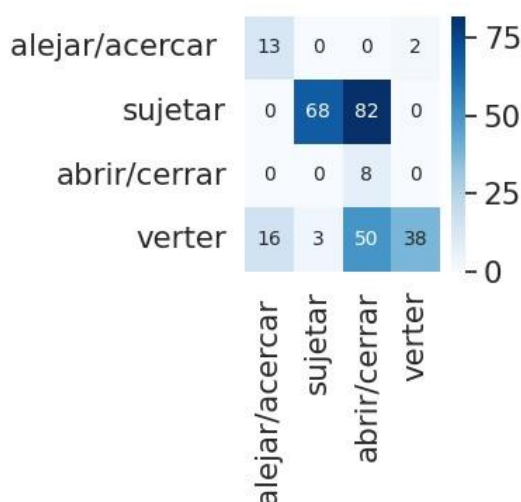


Figura 31. Matriz de confusión de la Tabla 10

Por último, el movimiento alejar/acercar tiene una sensibilidad como especificidades altas, no obstante, su precisión es baja, lo que quiere decir que el modelo tiene una alta tasa de falsos positivos. En la matriz de confusión vemos que 29 ventanas han sido clasificados como alejar/acercar, cuando solo 13 de ellos lo son.

Los resultados mostrados pertenecen a la red del lado derecho, el modelo del lado izquierdo presentaba resultados muy similares.

Con el objetivo de mejorar el modelo y solventar el problema con el *overfitting*, modificamos el tamaño de la ventana, a 15 con un solapamiento de 10, obteniendo más datos con los que entrenar el modelo y evitar el sobre entrenamiento.

II. Experimento 1.2: utilización del tamaño de ventana 15.

Probamos con el tamaño de ventana 15 en el lado izquierdo, con la red neuronal anterior, con 4 capas *Dense* de 32, 64 128 y 32 capas de entrada, con la función de activación "*relu*", en este caso contamos con 20.964 parámetros, mostrada la arquitectura de la red en la Figura 32. Observamos como a medida que transcurre el entrenamiento se produce una mejorar constante, indicando que el modelo está aprendiendo y ajustándose a los datos de *train*.

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
flatten (Flatten)           (None, 195)                 0
dense (Dense)                (None, 32)                  6272
dense_1 (Dense)              (None, 64)                  2112
dense_2 (Dense)              (None, 128)                 8320
dense_3 (Dense)              (None, 32)                  4128
dense_4 (Dense)              (None, 4)                   132
-----
Total params: 20,964
Trainable params: 20,964
Non-trainable params: 0
-----

```

Figura 32. Red neuronal de 4 capas Dense de 32, 64, 128 y 32 neuronas.

Aunque el *loss* y la *accuracy* del set de validación, muestran fluctuaciones y no mejoran de manera constante según avanza el entrenamiento. La *accuracy* del set de validación varía entre 0.287 y 0.695, lo que puede significar que el modelo no generaliza bien a los datos de validación.

Además, en algunas etapas, la precisión del set de validación es menor que la precisión de entrenamiento, lo cual puede ser otro indicio de *overfitting*. La pérdida de validación también es alta, es decir, el modelo tiene dificultades para predecir con precisión las salidas del conjunto de validación.

La discrepancia entre los resultados de entrenamiento y validación sugiere que es probable que se esté produciendo *overfitting*. Para confirmarlo, observamos el conjunto de *test* con una pérdida de 0.294, siendo relativamente baja, lo cual indica que el modelo está haciendo predicciones cercanas a las salidas reales en el conjunto de *test*. En cuanto a la precisión es de 0.915, lo que significa que el modelo clasifica bien las muestras.

Comparando estos resultados con los obtenidos en el conjunto de entrenamiento y validación, podemos ver que el rendimiento en el conjunto de prueba es similar o incluso mejor que el rendimiento en los otros conjuntos. Sugiere que el modelo generaliza bien a nuevos datos como los del set de *test*, pero no lo consigue con los del set de validación.

Probamos con la red neuronal modificada con la última capa de 256 neuronas, para ver cómo evoluciona el rendimiento de la red con más parámetros entrenables, un total de 50.756, presentada la arquitectura de la red en la Figura 33.

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
flatten (Flatten)           (None, 195)                 0
dense (Dense)                (None, 32)                  6272
dense_1 (Dense)              (None, 64)                  2112
dense_2 (Dense)              (None, 128)                 8320
dense_3 (Dense)              (None, 256)                 33024
dense_4 (Dense)              (None, 4)                   1028
-----
Total params: 50,756
Trainable params: 50,756
Non-trainable params: 0
-----

```

Figura 33. Red neuronal de 4 capas Dense de 32, 64, 128 y 256 neuronas.

En este caso, el entrenamiento se para en la época 19, con un *loss* de $3.577e-6$ y una *accuracy* de 1.00, mientras que el set de validación ha conseguido un *loss* de 1.419 y una *accuracy* de 0.788.

Señalando cierto sobre entrenamiento, para confirmarlo, observamos los resultados del set de *test*, teniendo un *loss* de 2.049 y una *accuracy* de 0.707. Podemos observar los resultados completos en la Tabla 11.

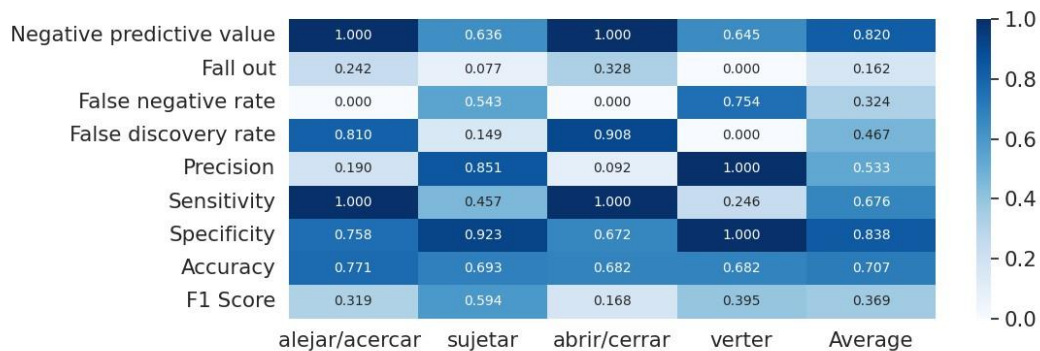


Tabla 11. Métricas de rendimiento con ventana de 15 frames.

Observamos la etiqueta Alejar/Acercar presenta una sensibilidad alta y una especificidad sutilmente elevada, junto con una precisión muy baja. Lo que quiere decir que tiene capacidad para detectar los casos positivos y tiene cierta capacidad para identificar los casos negativos y evitar falsos positivos.

Por el contrario, la precisión baja significa que la red tiene dificultades para distinguir correctamente entre los resultados positivos verdaderos y los falsos positivos.

Podemos verlo reflejado en la matriz de confusión de la Figura 34, la red ha clasificado 79 ventanas como Alejar/Acercar, cuando solamente 15 lo eran y 64 pertenecían al grupo de verter.

Lo mismo ocurre con Abrir/Cerrar, 98 ventanas han sido etiquetadas como tal, no obstante, solo 9 lo son, mientras que 75 de ellas son sujetar y 14 verter.

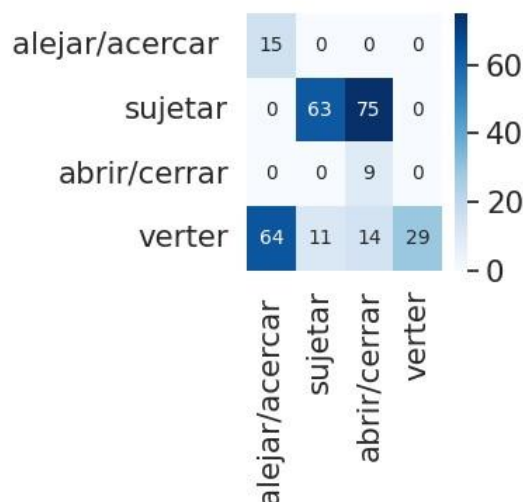


Figura 34. Matriz de confusión de la Tabla 11.

En las etiquetas sujetar y verter vemos una sensibilidad baja y una especificidad alta, con una precisión alta. En este caso la red tiene problemas para detectar los casos positivos, pero tiene buena capacidad para clasificar los casos negativos y positivos que se detectan. Aunque pierde algunos casos positivos, la gran mayoría de los resultados clasificados como positivos son realmente positivos, lo que resulta en una alta precisión.

Para verificarlo nos centramos en la matriz de confusión de la Figura 34 vemos que 29 ventanas se han identificado como verter, todas de manera correcta, sin embargo, 89 no han sido clasificadas como verter, siéndolo. Lo mismo ocurre con sujetar 74 han sido categorizadas como sujetar, de las cuales 63 realmente sí lo eran y 11 pertenecían a la clase verter, dejando sin clasificar como sujetar 75 ventanas.

Al ver los resultados obtenidos, llegamos a la conclusión de que la etiqueta de abrir y cerrar necesita ventanas de tamaño más grande para que pueda ser reconocida por la red.

III. Experimento 1.3: utilización del tamaño de ventana 45.

Tras realizar, el entrenamiento con ventanas de 15, aumentamos el tamaño de las ventanas hasta 45, con un solapamiento de 30. Generando ventanas de $\frac{3}{4}$ de segundo, con el objetivo de aumentar la cantidad de datos y salvar el sobreentrenamiento anterior.

Usamos el modelo secuencial desarrollado anteriormente con 6 capas *Dense*, con 32, 64, 128, 256, 64 y 32 capas de entrada, expuesto en la Figura 35. Además, cuenta con

varios *DropOut* entre las capas, un total de 5 con una desactivación de neuronas de 0.2. El modelo cuenta con 82.948 parámetros que entrenar.

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
flatten (Flatten)           (None, 650)                 0
dense (Dense)                (None, 32)                 20832
dropout (Dropout)           (None, 32)                 0
dense_1 (Dense)             (None, 64)                 2112
dropout_1 (Dropout)         (None, 64)                 0
dense_2 (Dense)             (None, 128)                8320
dropout_2 (Dropout)         (None, 128)                0
dense_3 (Dense)             (None, 256)               33024
dropout_3 (Dropout)         (None, 256)                0
dense_4 (Dense)             (None, 64)                 16448
dropout_4 (Dropout)         (None, 64)                 0
dense_5 (Dense)             (None, 32)                 2080
dense_6 (Dense)             (None, 4)                  132
-----
Total params: 82,948
Trainable params: 82,948
Non-trainable params: 0

```

Figura 35. Red neuronal con 6 capas Dense y con DropOut

Esta red comienza con una pérdida alta y una precisión baja, a medida que avanza el entrenamiento, pérdida disminuye gradualmente, lo que indica que la red está aprendiendo a ajustar sus pesos y realizar mejores predicciones, terminando el entrenamiento con un *loss* de 0.206. Así como la precisión que aumenta según la red aprende y se ajusta a los datos de entrenamiento, finalizando con una *accuracy* de 0.921.

En el set de validación, se observa una tendencia similar, la pérdida disminuye hasta 0.513 y la precisión aumenta hasta 0.841. En cuanto al conjunto de *test*, se obtiene una pérdida de 0.668 y una precisión de 0.879. Estos valores indican que el desempeño de la red es bueno, podemos observar las métricas de rendimiento en la Tabla 12.

Negative predictive value	0.915	1.000	1.000	0.797	0.928
Fall out	0.180	0.000	0.119	0.011	0.078
False negative rate	0.173	0.000	0.000	0.756	0.232
False discovery rate	0.331	0.000	0.542	0.120	0.248
Precision	0.669	1.000	0.458	0.880	0.752
Sensitivity	0.827	1.000	1.000	0.244	0.768
Specificity	0.820	1.000	0.881	0.989	0.922
Accuracy	0.822	1.000	0.892	0.803	0.879
F1 Score	0.740	1.000	0.629	0.383	0.688
	alejar/acercar	sujetar	abrir/cerrar	verter	Average

Tabla 12. Métricas de rendimiento de ventana de 45 frames.

Observamos que el movimiento Alejar/Acercar, tiene una sensibilidad y especificidad relativamente alta, muy similar. No obstante, presenta una precisión más baja. Nos apoyamos en la matriz de confusión de la Figura 36, vemos como la red ha predicho 136 ventanas como alejar/acercar, cuando verdaderamente lo eran 91 y 45 pertenecían a la clase verter. Por lo que indica que la red está generando falsos positivos.

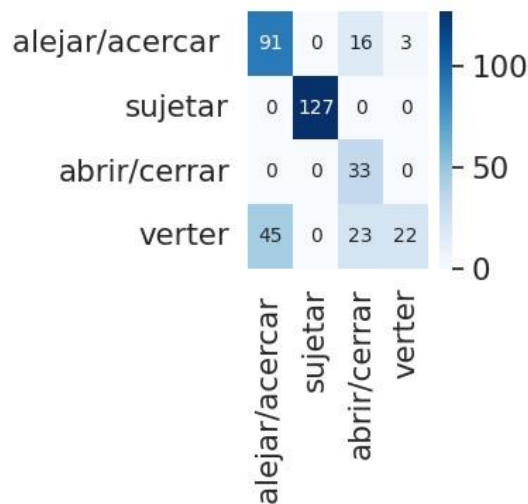


Figura 36. Matriz de confusión de la Tabla 12.

Con la etiqueta sujetar tenemos tanto una sensibilidad como una especificidad alta, junto con una precisión de 1, generando muy buenas predicciones. Esto se ve reflejado en la matriz de confusión de la Figura 36 con 127 ventanas clasificadas como sujetar de las cuales el 100% realmente lo son, generando ningún falso negativo.

La categoría de Abrir/Cerrar genera numerosos falsos positivos, lo vemos por su elevada sensibilidad y especificidad, y baja precisión. Vemos que ha clasificado como Abrir/Cerrar un total de 72, cuando realmente lo eran 33, 23 de ellas pertenecían a la clase verter y 16 a Alejar/Acercar.

Ocurre lo contrario con la clase de verter, tiene una sensibilidad baja y una especificidad alta, con una precisión bastante alta. Examinamos la matriz de confusión de la Figura 36 y vemos que solo 25 ventanas fueron clasificadas como verter, errando en 2 de ellas que eran alejar/acercar. Lo que señala que se produjeron 68 falsos positivos que realmente si eran verter.

En resumen, esta red confunde el movimiento de verter con otros y clasifica como abrir/cerrar movimientos que realmente no lo son.

Tras este entrenamiento, decidimos fijar el tamaño de ventana en 45 con un solapamiento de 30 para el resto de los entrenamientos. Debido a que este tamaño proporciona ventanas lo suficientemente grandes como para que la red diferencie

los movimientos y sea lo suficientemente pequeña como para que se generen las ventanas necesarias para entrenar correctamente la red.

Experimento 2: selección de sujetos sanos.

Procedimos a analizar en profundidad los vídeos de los sujetos sanos, con la finalidad de establecer un patrón de movimiento entre ellos y disponer grupos lo más heterogéneos posibles a la hora de asignar que sujetos pertenece al set de *train*, validación o *test*.

Podemos observar en la Tabla 13 como predomina el lado derecho, al ser la gran mayoría de sujetos diestros. Determinamos la presencia de 3 patrones, fijándonos en como realiza las actividades de abrir, verter y cerrar, tal y como se muestra en la Tabla 13 durante la ejecución del movimiento *Open/Close a bottle and pour water*:

- Patrón 1 (P1): la secuencia de movimientos comienza con la mano derecha sujetando la botella y con la izquierda abriéndola, vierten el agua con la mano derecha y terminan cerrando la botella con la mano izquierda.
- Patrón 2 (P2): la secuencia de movimientos comienza con la mano izquierda sujetando la botella y con la derecha abriéndola, vierten el agua con la mano izquierda y terminan cerrando la botella con la mano derecha.
- Patrón 3 (DCHA): la secuencia de movimientos comienza sujetando la botella con la mano izquierda y abriéndola con la mano derecha. Posteriormente, realizan una transferencia de la botella de la mano izquierda a la mano derecha y vierten el agua con la mano derecha. Por último, realiza otra transferencia de botella de la mano derecha a la mano izquierda, cerrando la botella con la mano derecha.

SUJETO	ACERCAR	TRANSF. 1	SUJETAR	ABRIR	TRANSF. 2	VERTER	TRANSF. 3	CERRAR	ALEJAR	PATRON
S01	2	NO	derecha	izquierda	NO	derecha	NO	izquierda	derecha	P1
S02	izquierda	NO	izquierda	derecha	NO	izquierda	NO	derecha	derecha	P2
S03	derecha	NO	derecha	izquierda	NO	derecha	NO	izquierda	derecha	P1
S04	derecha	NO	izquierda	derecha	NO	izquierda	NO	derecha	derecha	P2
S05	derecha	NO	derecha	izquierda	NO	derecha	NO	izquierda	derecha	P1
S06	izquierda	NO	izquierda	derecha	SI	derecha	SI	derecha	derecha	DCHA
S07	derecha	NO	derecha	izquierda	NO	derecha	NO	izquierda	derecha	P1
S08	derecha	NO	derecha	izquierda	NO	derecha	NO	izquierda	derecha	P1
S09	dcha-2	NO	izquierda	derecha	SI	derecha	SI	derecha	2	DCHA
S10	derecha	SI	izquierda	derecha	NO	izquierda	NO	derecha	derecha	P2
S12	derecha	NO	izquierda	derecha	SI	derecha	SI	derecha	derecha	DCHA
S13	derecha	NO	izquierda	derecha	NO	izquierda	NO	derecha	derecha	P2
S14	2	NO	derecha	izquierda	NO	derecha	NO	izquierda	2	P1
S15	izquierda	NO	derecha	izquierda	NO	derecha	NO	izquierda	izquierda	P1

Tabla 13. Análisis de patrones de movimiento de sujetos adultos sanos.

Realizamos la selección de sujetos de tal forma que haya representación de todos los patrones en todos los grupos: entrenamiento, validación y *test*.

Comenzamos con el lado derecho, estableciendo en el set de entrenamiento a los sujetos: S02, S03, S06, S07, S08, S09, S12, S13 y S14; en el set de validación a S01 y S04; y en el set de *test* a S05, S10 y S15.

En el lado izquierdo establecemos en el set de entrenamiento a los sujetos: S01, S02, S04, S05, S06, S07, S09, S12, S14; en el set de validación a S03 y S08; y en el set de *test* a S10, S13 y S15.

Experimento 3: empleo de arquitectura “cono invertido” con redes Dense.

Realizamos una prueba empleando una estructura de “cono invertido”, es decir, comenzamos con una capa con pocas neuronas y vamos aumentándolas hasta una red, después la siguiente va decreciendo el tamaño hasta la final que vuelve al tamaño de la inicial.

En el caso de esta red, cuenta con un total de 7 capas *Dense* de 32, 64, 128, 256, 128, 64, 32, intercalándolas con *DropOuts* con una desactivación de 0.2. El modelo cuenta con un total de 105.572 parámetros entrenables, como podemos observar en la Figura 37.

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
flatten (Flatten)           (None, 585)                 0
dense (Dense)                (None, 32)                 18752
dropout (Dropout)           (None, 32)                 0
dense_1 (Dense)              (None, 64)                 2112
dropout_1 (Dropout)         (None, 64)                 0
dense_2 (Dense)              (None, 128)                8320
dropout_2 (Dropout)         (None, 128)                0
dense_3 (Dense)              (None, 256)                33024
dropout_3 (Dropout)         (None, 256)                0
dense_4 (Dense)              (None, 128)                32896
dropout_4 (Dropout)         (None, 128)                0
dense_5 (Dense)              (None, 64)                 8256
dropout_5 (Dropout)         (None, 64)                 0
dense_6 (Dense)              (None, 32)                 2080
dense_7 (Dense)              (None, 4)                  132
-----
Total params: 105,572
Trainable params: 105,572
Non-trainable params: 0

```

Figura 37. Red neuronal con 7 capas Dense y con Dropout, imitando arquitectura de “cono invertido”

Obtenemos un *loss* de 0.115 y una *accuracy* de 0.955 en el set de entrenamiento y un *loss* de 0.738 y una *accuracy* de 0.791 en el set de validación. Además, si nos fijamos

en el set de *test*, alcanza un *loss* de 1.290 y una *accuracy* de 0.772. Los resultados mejoran, pero el *loss* sigue siendo alto.

6.1.2.2 ENTRENAMIENTO CON REDES CONVOLUCIONALES CON SUJETOS CONTROLES.

Experimento 4: utilización de redes Convolucionales simples

Como prueba para evitar ese sobreajuste al final del entrenamiento, probamos a entrenar otro tipo de red neuronal más compleja, una red convolucional, con los mismos sujetos.

Comenzamos usando una red neuronal convolucional típica, con capas de convolución seguidas de capas de *Pooling*, reutilizando la idea de los anteriores entrenamientos. Empleamos 3 capas de convolución con 32, 64 y 128 filtros y como función de activación "*relu*" y 3 capas de *Pooling*, con el objetivo de reducir el tamaño y disminuir la cantidad de parámetros.

Finaliza el modelo con una capa de agrupación *Dense* con 4 neuronas y la función de activación "*relu*" y con una capa de salida *Dense* con la función de activación de "*softmax*". Podemos observar el modelo anteriormente descrito en la Figura 38.

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)              (None, 45, 13, 32)         320
max_pooling2d (MaxPooling2D) (None, 23, 7, 32)          0
conv2d_1 (Conv2D)            (None, 23, 7, 64)          32832
max_pooling2d_1 (MaxPooling2D) (None, 12, 4, 64)          0
conv2d_2 (Conv2D)            (None, 12, 4, 128)         131200
max_pooling2d_2 (MaxPooling2D) (None, 6, 2, 128)          0
max_pooling2d_3 (MaxPooling2D) (None, 3, 1, 128)          0
dropout (Dropout)           (None, 3, 1, 128)          0
flatten (Flatten)            (None, 384)                 0
dense (Dense)                (None, 4)                   1540
-----
Total params: 165,892
Trainable params: 165,892
Non-trainable params: 0

```

Figura 38. Red neuronal convolucional con 4 capas y con DropOut.

En cuanto al entrenamiento, conseguimos un *loss* de 0.001 y una *accuracy* de 1.000 en el set de *train*, mientras que, en el set de validación, se muestra un *loss* de 0.657 y una *accuracy* de 0.900, siendo muy buen resultado. Sin embargo, en el set de *test*,

obtenemos un *loss* muy alto de 3.111 y una *accuracy* de 0.800. El entrenamiento de la red mejora, no obstante, el error que presenta continúa siendo alto.

Experimento 5: introducción de técnicas de regularización.

I. Experimento 5.1: empleo de Batch Normalization

Probamos a utilizar técnicas de regularización, introduciendo entre las capas convolucionales, una capa de *Batch Normalization*. Finalizando el modelo con un total de 166.068 parámetros entrenables como podemos observar en la Figura 39.

El entrenamiento finaliza en la época 16, mostrando un *loss* de 0.001 y una *accuracy* de 1.000 en el set de *train*, y un *loss* de 0.953 y una *accuracy* de 0.787 en el set de validación. Siendo resultados muy optimistas, sin embargo, si nos fijamos en el set de *test*, visualizamos un *loss* de 2.148 reduciéndose respecto a otros entrenamientos, pero siendo alto todavía. Además, obtenemos una *accuracy* de 0.791.

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)              (None, 45, 13, 32)         320
max_pooling2d (MaxPooling2D) (None, 23, 7, 32)          0
batch_normalization (BatchN (None, 23, 7, 32)          92
ormalization)
conv2d_1 (Conv2D)            (None, 23, 7, 64)          32832
max_pooling2d_1 (MaxPooling (None, 12, 4, 64)          0
2D)
batch_normalization_1 (Batc (None, 12, 4, 64)          48
hNormalization)
conv2d_2 (Conv2D)            (None, 12, 4, 128)         131200
max_pooling2d_2 (MaxPooling (None, 6, 2, 128)          0
2D)
batch_normalization_2 (Batc (None, 6, 2, 128)          24
hNormalization)
max_pooling2d_3 (MaxPooling (None, 3, 1, 128)          0
2D)
batch_normalization_3 (Batc (None, 3, 1, 128)          12
hNormalization)
dropout (Dropout)            (None, 3, 1, 128)          0
flatten (Flatten)            (None, 384)                 0
dense (Dense)                 (None, 4)                   1540
-----
Total params: 166,068
Trainable params: 165,980
Non-trainable params: 88

```

Figura 39. Red neuronal convolucional empleando DropOut y BatchNormalization

II. Experimento 5.2: empleo de Regularización L1 y L2

Posteriormente, probamos a incluir otra técnica de regularización, con la finalidad de mejorar el entrenamiento y aumentar la complejidad de la red. En este caso añadiríamos capas de las técnicas de regularización L1 y L2 intercaladas entre capas.

Utilizamos una red neuronal convolucional compuesta por 4 capas convolucionales, con 16, 32, 64 y 128 filtros. Formada, también, por 4 capas de *MaxPooling*, para reducir las dimensiones.

Además, compuesta por las técnicas de regularización introducidas anteriormente como *Batch Normalization* en cada salida de las capas convolucionales, para normalizar los valores de salida y acelerar el entrenamiento. También, incluimos varios *DropOut* para regularizar y evitar el sobreajuste, apagando aleatoriamente el 20% de las neuronas durante el entrenamiento.

Por último, para regularizar los datos insertamos las técnicas de regularización L2 y L1 en las capas convolucionales. Conseguimos el objetivo de aumentar la complejidad del modelo, incrementando el número de parámetros hasta 222.388, como podemos observar en la Figura 40.

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)              (None, 50, 13, 16)         160
max_pooling2d (MaxPooling2D) (None, 25, 7, 16)         0
batch_normalization (BatchN  (None, 25, 7, 16)         100
ormalization)
conv2d_1 (Conv2D)            (None, 25, 7, 32)          8224
max_pooling2d_1 (MaxPooling  (None, 13, 4, 32)          0
2D)
batch_normalization_1 (Batc  (None, 13, 4, 32)          52
hNormalization)
conv2d_2 (Conv2D)            (None, 13, 4, 64)          32832
max_pooling2d_2 (MaxPooling  (None, 7, 2, 64)          0
2D)
batch_normalization_2 (Batc  (None, 7, 2, 64)          28
hNormalization)
conv2d_3 (Conv2D)            (None, 7, 2, 128)          131200
max_pooling2d_3 (MaxPooling  (None, 4, 1, 128)         0
2D)
batch_normalization_3 (Batc  (None, 4, 1, 128)         16
hNormalization)
dropout (Dropout)            (None, 4, 1, 128)         0
flatten (Flatten)            (None, 512)                0
dense (Dense)                 (None, 128)                65664
dense_1 (Dense)               (None, 4)                  516
-----
Total params: 238,792
Trainable params: 238,694
Non-trainable params: 98

```

Figura 40. Red neuronal convolucional usando *DropOut*, *BatchNormalization* y regularización L1 y L2.

En general, podemos decir que este modelo muestra un buen rendimiento en términos de precisión, ya que alcanza valores superiores al 90% en varias etapas del entrenamiento tanto en el conjunto de entrenamiento como en el de validación.

Terminando el entrenamiento con una *accuracy* de 0.993 en el set de *train* y un 0.904 en el set de validación.

Si observamos los valores de la función de pérdidas, el modelo finaliza el entrenamiento con un *loss* de 0.103 en el set entrenamiento y 0.600 en el set de validación. Esto nos señala que el modelo está aprendiendo a ajustarse mejor a los datos de entrenamiento, siendo muy positivo.

Aunque, es importante resaltar que los valores de *loss* en el conjunto de validación son generalmente más altos que en el conjunto de *train*. Esto sugiere que el modelo puede estar sufriendo cierto sobreajuste, pero los resultados siguen siendo buenos.

Fijándonos en los resultados del conjunto de *test*, presenta una *accuracy* de 0.780 y un *loss* de 1.339, podemos ver que los resultados son peores, en comparación a los demás grupos, indicando que el modelo no está generalizando de manera efectiva a nuevos datos. El *loss* es relativamente alto, señalando que el modelo está teniendo dificultades para ajustarse a los datos de *test*.

Experimento 6: empleo de arquitectura “cono invertido” con redes Convolucionales.

Probamos a aumentar la complejidad del modelo, añadiendo una capa convolucional más con 256 neuronas. Igualmente, imitamos la estructura de “cono invertido” como lo realizamos en las capas *Dense*, aplicándolo a nuestro modelo de red neuronal convolucional.

En este caso, después de la capa de 256 neuronas, introducimos otras dos capas convolucionales, de 128 y 64 neuronas, tal y como se muestra en la Figura 41.

Model: "sequential"					
Layer (type)	Output Shape	Param #			
conv2d (Conv2D)	(None, 45, 13, 16)	160	conv2d_4 (Conv2D)	(None, 3, 1, 256)	524544
max_pooling2d (MaxPooling2D)	(None, 23, 7, 16)	0	max_pooling2d_4 (MaxPooling2D)	(None, 2, 1, 256)	0
batch_normalization (Batch Normalization)	(None, 23, 7, 16)	92	batch_normalization_4 (Batch Normalization)	(None, 2, 1, 256)	8
dropout (Dropout)	(None, 23, 7, 16)	0	dropout_4 (Dropout)	(None, 2, 1, 256)	0
conv2d_1 (Conv2D)	(None, 23, 7, 32)	8224	conv2d_5 (Conv2D)	(None, 2, 1, 128)	524416
max_pooling2d_1 (MaxPooling2D)	(None, 12, 4, 32)	0	max_pooling2d_5 (MaxPooling2D)	(None, 1, 1, 128)	0
batch_normalization_1 (Batch Normalization)	(None, 12, 4, 32)	48	batch_normalization_5 (Batch Normalization)	(None, 1, 1, 128)	4
dropout_1 (Dropout)	(None, 12, 4, 32)	0	dropout_5 (Dropout)	(None, 1, 1, 128)	0
conv2d_2 (Conv2D)	(None, 12, 4, 64)	32832	conv2d_6 (Conv2D)	(None, 1, 1, 64)	131136
max_pooling2d_2 (MaxPooling2D)	(None, 6, 2, 64)	0	max_pooling2d_6 (MaxPooling2D)	(None, 1, 1, 64)	0
batch_normalization_2 (Batch Normalization)	(None, 6, 2, 64)	24	batch_normalization_6 (Batch Normalization)	(None, 1, 1, 64)	4
dropout_2 (Dropout)	(None, 6, 2, 64)	0	dropout_6 (Dropout)	(None, 1, 1, 64)	0
conv2d_3 (Conv2D)	(None, 6, 2, 128)	131200	flatten (Flatten)	(None, 64)	0
max_pooling2d_3 (MaxPooling2D)	(None, 3, 1, 128)	0	dense (Dense)	(None, 64)	4160
batch_normalization_3 (Batch Normalization)	(None, 3, 1, 128)	12	dense_1 (Dense)	(None, 4)	260
dropout_3 (Dropout)	(None, 3, 1, 128)	0			
			Total params: 1,357,124		
			Trainable params: 1,357,028		
			Non-trainable params: 96		

Figura 41. Red neuronal convolucional imitando la arquitectura de “cono invertido”

El entrenamiento de esta red se para en la época 31, mostrándonos un *loss* de 0.735 y una *accuracy* de 0.770 en el set de entrenamiento. Comparándolo con el set de validación, que presenta un *loss* de 0.846 y una *accuracy* de 0.908, siendo resultados muy positivos y mejorando claramente el rendimiento de la red.

Si nos fijamos en el set de *test*, obtenemos un *loss* de 1.067 y una *accuracy* de 0.741, mostrándonos un buen rendimiento de la red, por lo que mantendremos la estructura de “cono invertido” para el resto de los entrenamientos.

Experimento 7: empleo de capa de Global Average Pooling (GAP).

I. Experimento 7.1: capa GAP y capa Dense de 32.

Para reducir la dimensionalidad de la red neuronal añadimos una capa de *Global Average Pooling*, muy utilizado en las redes convolucionales, como la que estamos empleando. Esta capa calcula el promedio de cada canal de la última capa convolucional y genera una representación de características unidimensional.

También para finalizar usamos dos capas *Dense* con 32 y 4 de entradas, con las funciones de activación “*relu*” y “*softmax*”. Podemos visualizar el modelo anteriormente comentado en la Figura 42.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 45, 13, 16)	160
max_pooling2d (MaxPooling2D)	(None, 23, 7, 16)	0
batch_normalization (Batch Normalization)	(None, 23, 7, 16)	92
dropout (Dropout)	(None, 23, 7, 16)	0
conv2d_1 (Conv2D)	(None, 23, 7, 32)	8224
max_pooling2d_1 (MaxPooling2D)	(None, 12, 4, 32)	0
batch_normalization_1 (Batch Normalization)	(None, 12, 4, 32)	48
dropout_1 (Dropout)	(None, 12, 4, 32)	0
conv2d_2 (Conv2D)	(None, 12, 4, 64)	32832
max_pooling2d_2 (MaxPooling2D)	(None, 6, 2, 64)	0
batch_normalization_2 (Batch Normalization)	(None, 6, 2, 64)	24
dropout_2 (Dropout)	(None, 6, 2, 64)	0
conv2d_3 (Conv2D)	(None, 6, 2, 128)	131200
max_pooling2d_3 (MaxPooling2D)	(None, 3, 1, 128)	0
batch_normalization_3 (Batch Normalization)	(None, 3, 1, 128)	12
dropout_3 (Dropout)	(None, 3, 1, 128)	0
conv2d_4 (Conv2D)	(None, 3, 1, 256)	524544
max_pooling2d_4 (MaxPooling2D)	(None, 2, 1, 256)	0
batch_normalization_4 (Batch Normalization)	(None, 2, 1, 256)	8
dropout_4 (Dropout)	(None, 2, 1, 256)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 256)	0
dense (Dense)	(None, 32)	8224
dense_1 (Dense)	(None, 4)	132

Total params: 705,500
 Trainable params: 705,408
 Non-trainable params: 92

Figura 42. Red neuronal convolucional empleando Global Average Pooling.

Continuamos probando con el lado derecho y los sujetos seleccionados anteriormente. Ampliamos el número de épocas hasta 50, sin embargo, el entrenamiento es interrumpido en la época 48.

Nos muestra un *loss* de 0.112 y una *accuracy* de 0.995 en el set de entrenamiento y un *loss* de 0.619 y una *accuracy* de 0.908 del set de validación. Mientras que en el set

de *test* hemos conseguido un *loss* de 2.280 y una *accuracy* de 0.791. El *loss* sigue siendo alto y puede mejorar, aunque hemos obtenido una precisión más alta.

II. Experimento 7.2: capa GAP y capa Dense de 64

Para tratar de mejorar la red aumentamos la dimensión de una de las capas finales *Dense* hasta 64. En este caso, el entrenamiento es interrumpido en la época 44.

Los resultados del conjunto de entrenamiento son similares al anterior, *loss* de 0.099 y una *accuracy* de 0.999, mientras que en el set de validación hemos alcanzado un *loss* de 0.843 y una *accuracy* de 0.879, sutilmente más bajo que en el modelo anterior.

No obstante, en el conjunto de *test*, hemos adquirido mejores resultados, disminuyendo el *loss* hasta 1.623 y la *accuracy* ha aumentado hasta 0.838. La Tabla 14 muestra las métricas de rendimiento del modelo, reflejando una gran mejora.

	alejar/acercar	sujetar	abrir/cerrar	verter	Average
Negative predictive value	0.910	1.000	1.000	0.938	0.962
Fall out	0.070	0.000	0.049	0.027	0.036
False negative rate	0.217	0.000	0.000	0.204	0.105
False discovery rate	0.174	0.000	0.309	0.098	0.145
Precision	0.826	1.000	0.691	0.902	0.855
Sensitivity	0.783	1.000	1.000	0.796	0.895
Specificity	0.930	1.000	0.951	0.973	0.964
Accuracy	0.887	1.000	0.956	0.930	0.943
F1 Score	0.804	1.000	0.817	0.846	0.867

Tabla 14. Métricas de rendimiento de red neurona convolucional con capa de Global Average Pooling.

En la etiqueta alejar/acercar tenemos alta sensibilidad y especificidad, así como una alta precisión, siguen produciéndose falsos positivos, no obstante, estos han disminuido. Podemos ver en la matriz de confusión de la Figura 43, que de 109 ventanas categorizadas como alejar/acercar, 90 realmente sí que lo eran y 19, eran verter.

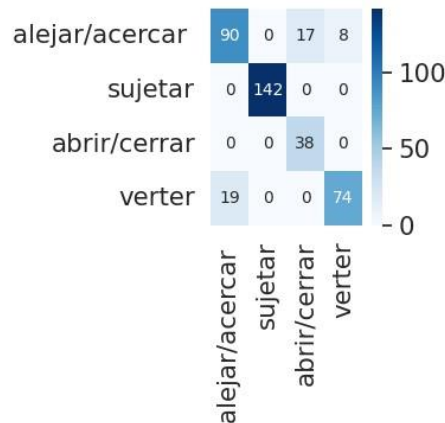


Figura 43. Matriz de confusión de la Tabla 14.

La categoría de sujetar sigue dando muy buenos resultados, 142 ventanas clasificadas como sujetar, en realidad si lo eran.

Para la etiqueta de abrir/cerrar, la sensibilidad y especificidad han aumentado, sin embargo, la precisión es algo baja en comparación ellas. Esto se ve reflejado en una tasa de falsos positivos, vemos que 55 ventanas se clasificaron como abrir/cerrar cuando 38 sí que lo eran y 17 eran alejar/acercar.

Por último, verter tiene una sensibilidad sutilmente más baja en comparación con su especificidad, pero su precisión sí que es buena. La matriz de confusión de la Figura 43 nos indica que 82 fueron clasificados como verter y solo erro en 8 de ellas.

Asimismo, se destaca 19 falsos negativos que fueron clasificados como alejar/acercar, que realmente eran verter, como hemos mencionado anteriormente.

III. Experimento 7.3: capa GAP y capa Dense de 128.

Con la finalidad de ver si el modelo sigue mejorando, aumentando la capa *Dense*, volvemos a aumentar su entrada hasta 128 y entrenamos la red. En esta ocasión, el entrenamiento se para en la época 32, además el entrenamiento no ha sido como esperábamos.

Los resultados del set de entrenamiento han sido muy similares que los dos anteriores, con un *loss* de 0.167 y una *accuracy* de 0.995, y en el set de validación también, con un *loss* de 0.875 y una *accuracy* de 0.816.

En cambio, en el set de *test*, los resultados han empeorado sutilmente, manteniendo un *loss* muy similar al anterior, pero empeorando la precisión, bajando hasta 0.791. Por lo que descartamos la idea de añadir una capa *Dense* de 128 de entrada.

6.1.2.3 ENTRENAMIENTO CON REDES CONVOLUCIONALES SIN ÁNGULOS DE LAS LUMBARES CON SUJETOS CONTROLES.

Experimento 8: lado derecho sin ángulos de las lumbares.

En los entrenamientos previos habíamos considerado los ángulos de "arm_flex_r", "arm_add_r", "arm_rot_r", "elbow_flex_r", "pro_sup_r", "arm_flex_l", "arm_add_l", "arm_rot_l", "elbow_flex_l", "pro_sup_l", "lumbar_extension", "lumbar_bending", "lumbar_rotation".

Sin embargo, tras numerosos entrenamientos probamos la idea de quitar los ángulos de "lumbar_extension", "lumbar_bending" y "lumbar_rotation", ya que algunos sujetos si se inclinaban para coger la botella y el vaso, pero otros no flexionaban la espalda a la hora de realizar el movimiento.

Entrenamos el modelo anteriormente utilizado con 7 capas convolucionales y *MaxPooling* juntos con la capa de *Global Average Pooling*, y las dos capas finales *Dense* de 64 y 4, que nos han dado buenos resultados. Añadiendo las técnicas de regularización: *DropOut* con una tasa de desactivación de neuronas de 0.2 y las capas de *Batch Normalization*, entre las capas convolucionales. El modelo consta de 713.852 parámetros en total.

El entrenamiento se para en la época 30, obteniendo un *loss* de 0.174 y una *accuracy* de 0.985 en el set de entrenamiento y en el set de validación un *loss* de 1.3673 y una *accuracy* de 0.7077, lo que puede indicar cierto sobreentrenamiento.

No obstante, si nos fijamos en el set de *test*, podemos ver cómo ha mejorado la red obteniendo un *loss* de 1.514 y una *accuracy* de 0.824, siendo resultados muy positivos para nuestra red.

Después de ver los resultados obtenidos sin los ángulos de la espalda decidimos seguir los entrenamientos sin dichos ángulos y solo considerar los siguientes ángulos: "arm_flex_r", "arm_add_r", "arm_rot_r", "elbow_flex_r", "pro_sup_r", "arm_flex_l", "arm_add_l", "arm_rot_l", "elbow_flex_l", "pro_sup_l".

I. Experimento 8.1: lado derecho utilizando Global Average Pooling con 2 capas Dense de 128 y 64.

Probamos a mejorar la red anterior añadiendo otra capa *Dense* al final, después del *Global Average Pooling*, quedando una capa *Dense* de 128 con la función de activación "relu", otra capa *Dense* de 64 con la función de activación "relu" y la capa final *Dense* de 4, con la función de activación "softmax". El modelo final cuenta con 738.556 parámetros finales.

Los resultados obtenidos son un *loss* de 0.194 y una *accuracy* de 0.993 en el set de *train* y en el set de validación, un *loss* de 1.294 y una *accuracy* de 0.603. Asimismo,

si nos fijamos en los resultados de *test*, tenemos un *loss* de 1.009 y una *accuracy* de 0.876, concluyendo el entrenamiento de este lado.

Experimento 9: lado izquierdo sin ángulos de las lumbares.

I. Experimento 9.1: capa GAP y 2 capas Dense de 128 y 64.

Una vez terminamos con el lado derecho probamos a entrenar el lado izquierdo, sin considerar los ángulos de la espalda con la red neuronal última que hemos utilizado.

El modelo cuenta con 5 capas convolucionales de 16, 32, 64, 128 y 256 filtros y 5 capas de *MaxPooling*. También está formado al final por una capa de *Global Average Pooling* y las dos capas *Dense* de 128 y 64 con la función de activación "*relu*" y una capa *Dense* final de 4 con la función de activación "*softmax*". Podemos visualizar la arquitectura del modelo final en la Figura 44.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 45, 13, 16)	160
max_pooling2d (MaxPooling2D)	(None, 23, 7, 16)	0
batch_normalization (Batch Normalization)	(None, 23, 7, 16)	92
dropout (Dropout)	(None, 23, 7, 16)	0
conv2d_1 (Conv2D)	(None, 23, 7, 32)	8224
max_pooling2d_1 (MaxPooling2D)	(None, 12, 4, 32)	0
batch_normalization_1 (Batch Normalization)	(None, 12, 4, 32)	48
dropout_1 (Dropout)	(None, 12, 4, 32)	0
conv2d_2 (Conv2D)	(None, 12, 4, 64)	32832
max_pooling2d_2 (MaxPooling2D)	(None, 6, 2, 64)	0
batch_normalization_2 (Batch Normalization)	(None, 6, 2, 64)	24
dropout_2 (Dropout)	(None, 6, 2, 64)	0
conv2d_3 (Conv2D)	(None, 6, 2, 128)	131200
max_pooling2d_3 (MaxPooling2D)	(None, 3, 1, 128)	0
batch_normalization_3 (Batch Normalization)	(None, 3, 1, 128)	12
dropout_3 (Dropout)	(None, 3, 1, 128)	0
conv2d_4 (Conv2D)	(None, 3, 1, 256)	524544
max_pooling2d_4 (MaxPooling2D)	(None, 2, 1, 256)	0
batch_normalization_4 (Batch Normalization)	(None, 2, 1, 256)	8
dropout_4 (Dropout)	(None, 2, 1, 256)	0
global_average_pooling2d (GlobalAveragePooling2D)	(None, 256)	0
dense (Dense)	(None, 128)	32896
dense_1 (Dense)	(None, 64)	8256
dense_2 (Dense)	(None, 4)	260
Total params: 738,556		
Trainable params: 738,464		
Non-trainable params: 92		

Figura 44. Red neuronal convolucional con 2 capas Dense finales de 128 y 64 neuronas.

Podemos observar en los resultados que incluso con el lado izquierdo, que consta de menos datos, obtenemos buenos resultados. En este caso, el entrenamiento se para en la época 22.

Los resultados del set de *train* tenemos un *loss* de 0,128 y una *accuracy* de 0.998 y en el set de validación un *loss* de 1.429 y una *accuracy* de 0.854. Podemos ver que la red tiene un rendimiento muy positivo apoyándonos en los resultados del set de *test*, con un *loss* de 0.735 y una *accuracy* de 0.806.

II. Experimento 9.2: capa GAP y 3 capas Dense de 128, 64 y 32.

Realizamos otra prueba añadiendo a la red anterior otras dos capas *Dense* tras la capa *Dense* de 128, una de 64 y otra de 32, todas ellas con la función de activación “*relu*”. Contando la red con un total de 740.972 parámetros.

Sin embargo, los resultados obtenidos son peores, en el set de *train* tenemos un *loss* de 0.144 y una *accuracy* de 0.994, en el set de validación un *loss* de 1.714 y una *accuracy* de 0.839. Si nos fijamos en el set de *test* vemos como el *loss* ha disminuido, pero al *accuracy* también, bajando hasta 0.832.

6.1.2.4 ENTRENAMIENTO CON REDES CONVOLUCIONALES CON SUJETOS PEDIÁTRICOS: SANOS Y PATOLÓGICOS.

Experimento 10: selección de sujetos pediátricos: sanos y patológicos.

Antes de comenzar a entrenar las redes con los sujetos pediátricos, tanto con los sujetos sanos como con los patológicos, procedemos a analizar en detalle los vídeos, tal y como realizamos con los sujetos controles. En la Tabla 15, podemos observar el estudio de los movimientos realizados por cada paciente.

SUJETO	ACERCAR	SUJETAR	ABRIR	VERTER	2 MANOS	CERRAR	ALEJAR
P01	2	Derecha	Izquierda	-	SI	Derecha	2
P02	2	Izquierda	Derecha	Izquierda	NO	Derecha	2
P03	2	Izquierda	Derecha	-	SI	Derecha	2
P04	No grab	Derecha	Izq-Dcha	-	SI	Izquierda	2
P05	No grab	Derecha	Izquierda	Izquierda	NO	Izquierda	2
P08	2	Izquierda	Derecha	-	SI	Derecha	Derecha
S11	Dcha-Izq	Izquierda	Derecha	-	SI	Derecha	Derecha
S16	2	Izquierda	Derecha	-	SI	Derecha	Derecha
S17	No grab	Derecha	Izq-Dcha	-	SI	Derecha	Derecha

Tabla 15. Análisis de patrones de movimiento de sujetos pediátricos.

Los sujetos correspondientes a P06 y P07, no pudieron ser incluidos en la base de datos, ya que se realizaron sus grabaciones, pero hubo problemas con las cámaras y el *software* no pudo reconstruir bien el esqueleto de los sujetos.

Los sujetos S11, S16 y S17 pertenecen a sujetos pediátricos sanos, corresponden a los hermanos mellizos de los sujetos pediátricos patológicos que se grabaron previamente.

Decidimos descartar el movimiento de acercar/alejar al haber sido realizado con las dos manos por los sujetos y al tener un problema con la sincronización de las cámaras, que impidió la grabación de esta etiqueta. Además, descartamos otra etiqueta por falta de datos, la categoría *verter*.

Igualmente, se incluyó una nueva etiqueta de verter con 2 manos como “2Manos”, debido a que los niños realizaban este movimiento con las dos manos y verter se considera como un movimiento realizado con una sola mano. Esto se debe a que los niños al tener las manos más pequeñas y la botella era de un tamaño grande, para poder sujetar correctamente la botella, la sujetaban con las dos manos, en cambio, los adultos con una mano pueden controlar bien el movimiento de verter agua.

Viendo el análisis en la Tabla 15 determinamos entrenar una red solo para el lado derecho, debido a la escasa presencia de datos del lado izquierdo, como podemos ver en la Tabla 16, que serían insuficientes para entrenar correctamente una red con ese lado.

SUJETO	PATOLOGÍA	LADO AFECTADO
P01	Hemiparesia	Derecho
P02	Hemiparesia	Izquierdo
P03	Hemiparesia	Izquierdo
P04	Hemiparesia	Derecho
P05	Diparesia	No afectación-diestro
P08	Hemiparesia	Izquierdo

Tabla 16. Patologías de los sujetos pediátricos

Experimento 11: arquitectura “cono invertido” con capa GAP y 2 capas Dense de 128 y 64 con sujetos pediátricos patológicos.

En primer lugar, comenzamos con la red convolucional desarrollada para los sujetos sanos, de 5 capas convolucionales de 16, 32, 64, 128 y 256 filtros y 5 capas de *MaxPooling*. Compuesto, también, por una capa de *Global Average Pooling* y las dos capas Dense de 128 y 64 con la función de activación “*relu*” y una capa Dense final de 3 con la función de activación “*softmax*”. El modelo final cuenta con un total de 738.491 parámetros.

En el caso de los sujetos patológicos, el entrenamiento se para en la época 12. Los resultados del set de *train* tenemos un *loss* de 0.379 y una *accuracy* de 0.996 y en el set de validación un *loss* de 8.609 y una *accuracy* de 0.411.

Podemos ver que la red no tiene un rendimiento muy bueno con el set de validación, en cambio en el set de *test*, tenemos un *loss* de 2.028 y una *accuracy* de 0.684. Se obtuvieron las métricas de rendimiento mostradas en la Tabla 17.

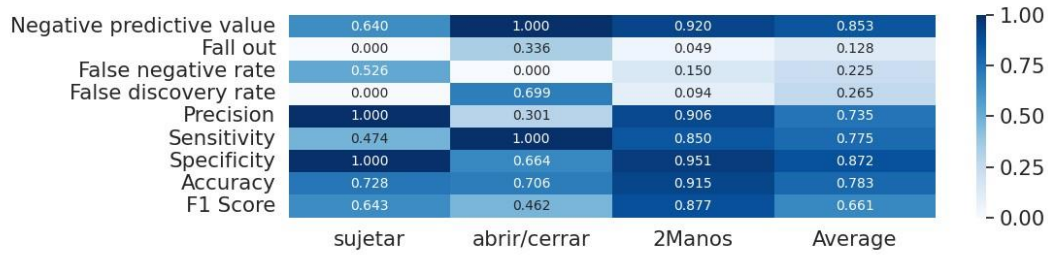


Tabla 17. Métricas de rendimiento de entrenamiento con sujetos patológicos.

Podemos observar como la sensibilidad de la etiqueta sujetar es baja y la especificidad muy alta junto con una precisión alta también. Lo vemos reflejado en la matriz de confusión de la Figura 45, 101 ventanas fueron clasificadas como sujetar, siendo todas ellas dicha etiqueta, no obstante, obtuvo 112 falsos negativos, confundiéndolas con las otras clases.

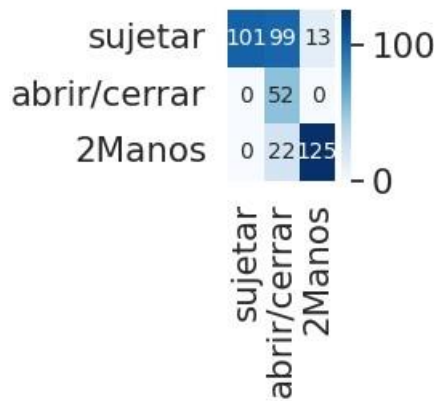


Figura 45. Matriz de confusión de la Tabla 17.

Ocurre lo contrario con la clase Abrir/Cerrar, presenta una sensibilidad alta y una especificidad baja con una precisión baja. Se muestra en la matriz de confusión de la Figura 45, 173 ventanas fueron clasificadas como Abrir/Cerrar, cuando solamente 52 sí que lo eran en realidad, confundiendo sobre todo con sujetar, un total de 99 casos.

La clase que mejores resultados proporciona es 2Manos, con un elevada sensibilidad, especificidad y precisión. En la matriz de confusión observamos que 138 ventanas fueron clasificadas como tal, equivocándose solamente en 13.

En resumen, la red reconoce bien la etiqueta de 2Manos, sin embargo, comete numerosos falsos positivos con la clase de abrir/cerrar.

Experimento 12: arquitectura “cono invertido” con capa GAP y 2 capas Dense de 128 y 64 con sujetos pediátricos sanos y patológicos.

Para aumentar la base de datos y mejorar el entrenamiento de la red, usamos los 3 sujetos pediátricos controles: S11, S16 y S17. Estos sujetos eran hermanos mellizos de los pacientes con parálisis, por lo que es interesante tenerlos en cuenta.

Nuevamente empleamos la red anterior con todos los sujetos pediátricos que disponemos, tanto los patológicos como lo no patológicos. La red se detuvo en la época 22.

Nos muestra un *loss* de 0.248 y una *accuracy* de 0.984 en el set de entrenamiento y en el set de validación obtenemos un *loss* de 6.514 y una *accuracy* de 0.507. Los resultados del set de validación son bajos al haber empleado un solo sujeto en dicho grupo, y así poder entrenar con más sujetos el modelo. Mientras en el set de *test*, hemos conseguido un *loss* de 1.628 y una *accuracy* de 0.745. Podemos ver los resultados más detallados reflejados en la Tabla 18.

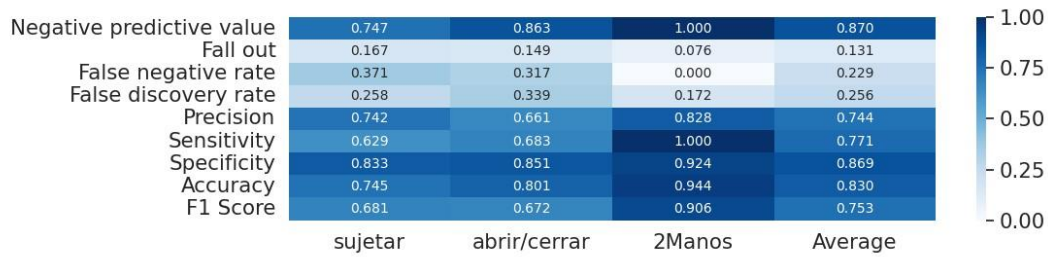


Tabla 18. Métricas de rendimiento de entrenamiento con sujetos pediátricos sanos y patológicos.

Apreciamos que la etiqueta sujetar ha empeorado sutilmente su rendimiento, disminuyendo tanto su sensibilidad y su especificidad, en la matriz de confusión de la Figura 46, vemos como 151 ventanas han sido clasificadas como dicha clase de las cuales, 39 fueron falsos positivos, además, obtuvo un total de 66 falsos negativos.

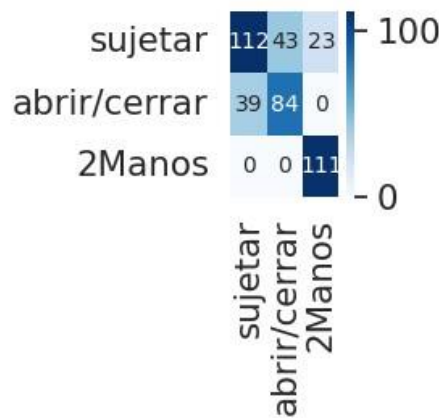


Figura 46. Matriz de confusión de la Tabla 18.

En cambio, la etiqueta de Abrir/Cerrar ha mejorado discretamente incrementando su especificidad y precisión, apreciamos que 127 ventanas fueron categorizadas como abrir/cerrar, de los cuales 43 fueron falsos positivos, también, tuvo 39 falsos negativos.

El rendimiento de la etiqueta 2Manos continuó siendo muy bueno, con alta sensibilidad, especificidad y precisión. Clasificadas como tal un total de 134 ventanas como 2Manos, siendo 23 falsos positivos.

En definitiva, la red reconoce bien el movimiento de 2Manos, teniendo ciertas dificultades con el resto de las categorías.

6.1.2.5 ENTRENAMIENTO SUJETOS CONTROLES Y PEDIÁTRICOS.

Experimento 13: entrenamiento con las etiquetas: sujetar, abrir/cerrar y 2Manos.

Probamos la red neuronal anteriormente mencionada, utilizando los sujetos pediátricos y los sujetos controles, repartiéndoles entre los diferentes sets de entrenamiento, validación y test.

Se asignó al set de entrenamiento a los sujetos: P01, P02, P03, S02, S06, S07, S08, S09, S10, S11, S12, S13, S14, S15, S16 y S17. Dentro del set de validación a S01 y S03 y en el set de *test* a P04, P05, P08, S04 y S05.

En este caso el entrenamiento duró hasta la época 31 con un *loss* de 0.180 y una *accuracy* de 0.973 en el set de entrenamiento, en cambio el set de validación presenta un *loss* de 0.320 y una *accuracy* de 0.816. Mejorando los resultados del set de *test*, ha disminuido el *loss* hasta 0.695 y ha aumentado la *accuracy* hasta 0.898.

En la Tabla 19 se muestra los resultados, siendo bastante positivos con la gran mayoría de los movimientos etiquetados.

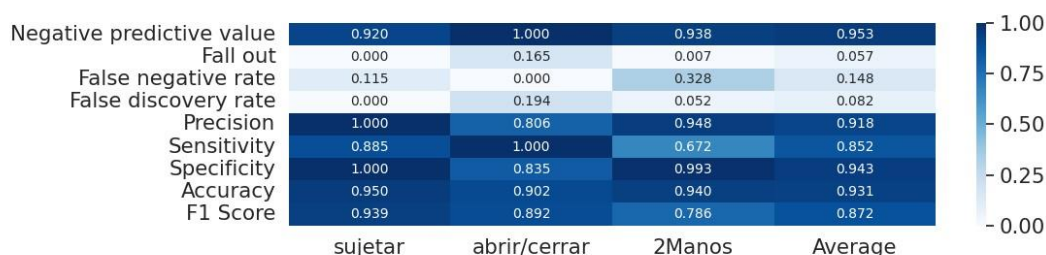


Tabla 19. Métricas de rendimiento de entrenamiento con sujetos sanos y patológicos.

Observamos que el movimiento sujetar ha mejorado con respecto a los sujetos pediátricos solos, aumentando la sensibilidad, especificidad y precisión. Vemos en la matriz de confusión de la Figura 47 que 314 ventanas fueron clasificadas correctamente como dicha categoría y solo obtuvo 41 falsos negativos.

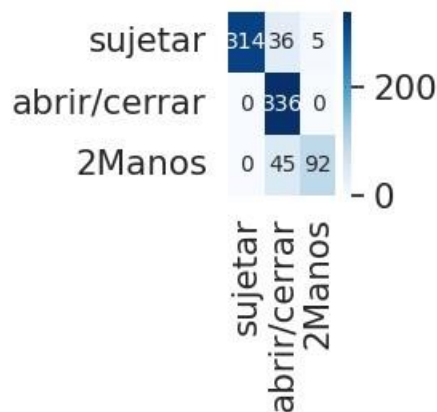


Figura 47. Matriz de confusión de la Tabla 19.

Igualmente, la clase abrir/cerrar presenta mejores métricas, clasificó 417 ventanas como dicha etiqueta, obteniendo solamente 81 falsos positivos.

El rendimiento de 2Manos se mantiene constante, con muy buenas estadísticas, mostrando 92 ventanas bien clasificadas, 5 falsos positivos y 45 falsos negativos.

Concluimos que se pueden realizar entrenamientos de forma conjunta entre los diferentes tipos de sujetos, obteniendo buenas métricas de rendimiento a la hora de clasificar los movimientos que componen el movimiento de *Open/Close a bottle and pour water*.

Experimento 14: añadimos la etiqueta de verter.

Probamos a introducir otras etiquetas como verter, presente en las capturas de los participantes sanos. Observamos un *loss* de 0.488 y una *accuracy* de 0.948 en el set de entrenamiento, mientras que en el set de validación contamos con un *loss* de 3.472 y una *accuracy* de 0.560.

No obstante, el set de *test* tiene mejores resultados con un *loss* de 1.089 y una *accuracy* de 0.855, siendo un resultado muy positivo, mostrados en la Tabla 20.

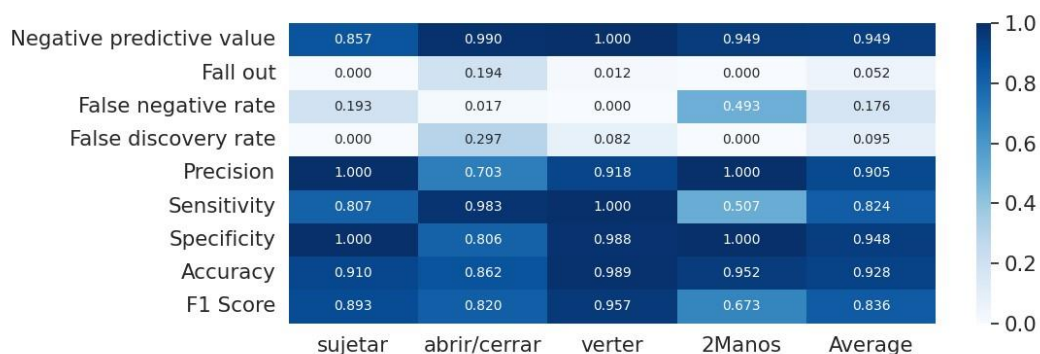


Tabla 20. Métricas de rendimiento con la etiqueta verter.

La clase sujetar y abrir/cerrar presentan métricas muy similares al anterior entrenamiento sin incluir la nueva clase.

Mientras que sujetar presenta ciertos falsos negativos, abrir/cerrar genera más falsos positivos, en este último caso sí que interfiere la categoría de 2Manos.

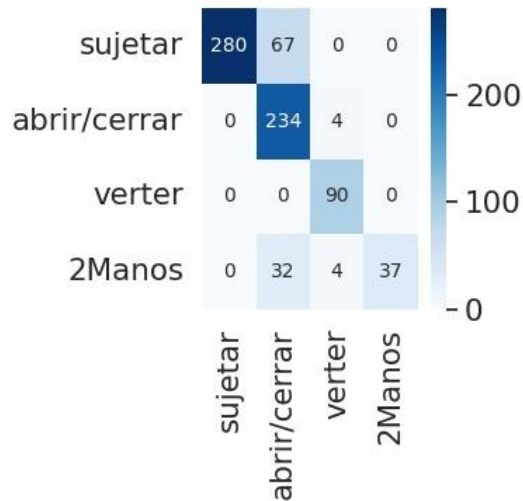


Figura 48. Matriz de confusión de la Tabla 20.

Observamos en la matriz de confusión de la Figura 48, que verter cuenta con buenos resultados, con una sensibilidad, especificidad y precisión alta, reconociendo todas las ventanas de esta clase y generando solamente 8 falsos positivos. Sin embargo, la sensibilidad de 2Manos se ha reducido discretamente, produciendo algunos falsos negativos, confundándose con abrir/cerrar.

Vemos que sí sería posible reconocer los dos movimientos relacionados con verter agua, siempre y cuando se creen dos etiquetas distintas para identificar el movimiento realizado con una mano o con dos manos.

Podemos observar un resumen de los experimentos obtenidos con los sujetos controles en la Tabla 21 y la Tabla 22. Así como, de los experimentos con los sujetos pediátricos y con todos los sujetos en la Tabla 23.

RESULTADOS BLOQUE II: SEGMENTACIÓN DE LAS FASES DEL MOVIMIENTO <i>OPEN/CLOSE A BOTTLE AND POUR WATER</i>												
Tipo de sujeto	Grupo al que pertenecen			Red neuronal					Resultados obtenidos			
	Set <i>train</i>	Set <i>valid.</i>	Set <i>test</i>	Tipo de arquitectura	Tam. de ventana	Nº de capas	Parámetros entrenab.	Técnicas de regular.	Parada (<i>epoch</i>)	Set de <i>train</i>	Set de <i>valid.</i>	Set de <i>test</i>
CONTROLES	"S01", "S02", "S03", "S04", "S05", "S06", "S07", "S08", "S09", "S12"	"S13" y "S14"	"S10" y "S15"	Dense	15	4 Dense: 32, 64, 128 y 256	56.996	NO	19	Loss: 3.577e-6 Accuracy: 1.00	Loss: 0.894 Accuracy: 0.805	Loss: 2.056 Accuracy: 0.875
					30		50.760	NO	11	Loss: 3.577e-6 Accuracy: 1.00	Loss: 1.419 Accuracy: 0.788	Loss: 2.049 Accuracy: 0.707
					45		82.948	DropOut	10	Loss: 0.206 Accuracy: 0.921	Loss: 0.513 Accuracy: 0.841	Loss: 0.668 Accuracy: 0.879
	"S02", "S03", "S06", "S07", "S08", "S09", "S12", "S13" y "S14"	"S01" y "S04"	"S05", "S10" y "S15"	Dense + "Cono invertido"	45	7 Dense: 32, 64, 128, 256, 128, 64, 32	105.575	DropOut	17	Loss: 0.115 Accuracy: 0.955	Loss: 0.738 Accuracy: 0.791	Loss: 1.290 Accuracy: 0.772
				CNN.		3 CNN: 32, 64, 128	165.892	DropOut	20	Loss: 0.001 Accuracy: 1.000	Loss: 0.657 Accuracy: 0.900	Loss: 3.111 Accuracy: 0.800
						3 CNN: 32, 64, 128	166.068	DropOut+ Batch Regular.	16	Loss: 0.001 Accuracy: 1.000	Loss: 0.953 Accuracy: 0.787	Loss: 2.148 Accuracy: 0.791
						4 CNN: 16, 32, 64, 128	238.792	DropOut+ Batch Regular. +	29	Loss: 0.103 Accuracy: 0.993	Loss: 0.600 Accuracy: 0.904	Loss: 1.339 Accuracy: 0.780
						CNN+ "Cono invertido"	4 CNN: 16, 32, 64, 128, 256, 128, 64	1.357.1214	Regular. L1 y L2	31	Loss: 0.735 Accuracy: 0.770	Loss: 0.846 Accuracy: 0.908

Tabla 21. Resumen I de los resultados de los experimentos del Bloque II: segmentación de las fases del movimiento *Open/Close a bottle and pour water*.

RESULTADOS BLOQUE II: SEGMENTACIÓN DE LAS FASES DEL MOVIMIENTO <i>OPEN/CLOSE A BOTTLE AND POUR WATER</i>													
Tipo de sujeto	Grupo al que pertenecen			Red neuronal					Resultados obtenidos				
	Set <i>train</i>	Set <i>valid.</i>	Set <i>test</i>	Tipo de arquitectura	Tam. de ventana	Nº de capas	Parámetros entrenab.	Técnicas de regular.	Parada (<i>epoch</i>)	Set de <i>train</i>	Set de <i>valid.</i>	Set de <i>test</i>	
CONTROLES	"S02", "S03", "S06", "S07", "S08", "S09", "S12", "S13" y "S14"	"S01" y "S04"	"S05", "S10" y "S15"	CNN+ "Cono invertido" + GAP	45	7 CNN: 16, 32, 64, 128, 256, 128, 64 + GAP + <i>Dense</i> 32	705.500	DropOut+ Batch Regular. + Regular. L1 y L2	48	Loss: 0.112 Accuracy: 0.995	Loss: 0.619 Accuracy: 0.908	Loss: 2.280 Accuracy: 0.791	
						7 CNN: 16, 32, 64, 128, 256, 128, 64 + GAP + <i>Dense</i> 32	713.852		44	Loss: 0.099 Accuracy: 0.999	Loss: 0.843 Accuracy: 0.879	Loss: 1.623 Accuracy: 0.838	
						7 CNN: 16, 32, 64, 128, 256, 128, 64 + GAP + <i>Dense</i> 32	730.556		32	Loss: 0.167 Accuracy: 0.995	Loss: 0.875 Accuracy: 0.816	Loss: 1.761 Accuracy: 0.791	
	ELIMINACIÓN DE LOS ÁNGULOS DE LA ESPALDA												
	"S02", "S03", "S06", "S07", "S08", "S09", "S12", "S13" y "S14"	"S01" y "S04"	"S05", "S10" y "S15"	CNN+ "Cono invertido" + GAP	45	7 CNN: 16, 32, 64, 128, 256, 128, 64 + GAP + 2 <i>Dense</i> 128, 64	738.556	DropOut+ Batch Regular. + Regular. L1 y L2	17	Loss: 0.194 Accuracy: 0.993	Loss: 1.294 Accuracy: 0.603	Loss: 1.009 Accuracy: 0.876	
						7 CNN: 16, 32, 64, 128, 256, 128, 64 + GAP + 2 <i>Dense</i> 128, 64	738.556		22	Loss: 0.128 Accuracy: 0.998	Loss: 1.429 Accuracy: 0.854	Loss: 0.735 Accuracy: 0.806	
						7 CNN: 16, 32, 64, 128, 256, 128, 64 + GAP + 3 <i>Dense</i> 128, 64, 32	740.972		16	Loss: 0.144 Accuracy: 0.994	Loss: 1.714 Accuracy: 0.839	Loss: 0.701 Accuracy: 0.832	

Tabla 22. Resumen II de los resultados de los experimentos del Bloque II: segmentación de las fases del movimiento *Open/Close a bottle and pour water*.

RESULTADOS BLOQUE II: SEGMENTACIÓN DE LAS FASES DEL MOVIMIENTO <i>OPEN/CLOSE A BOTTLE AND POUR WATER</i>												
Tipo de sujeto	Grupo al que pertenecen			Red neuronal					Resultados obtenidos			
	Set <i>train</i>	Set <i>valid.</i>	Set <i>test</i>	Tipo de arquitectura	Tam. de ventana	Nº de capas	Parámetros entrenab.	Técnicas de regular.	Parada (<i>epoch</i>)	Set de <i>train</i>	Set de <i>valid.</i>	Set de <i>test</i>
PEDIÁTRICOS PATOLÓGICOS	"P02", "P04", "P05", "P08"	"S16"	"P01", "P03"	CNN+ "Cono invertido" + GAP	45	5 CNN: 16, 32, 64, 128, 256 + GAP + <i>Dense</i> 128, 64	738.556	DropOut+ Batch Regular. + Regular. L1 y L2	11	Loss: 0.379 Accuracy: 0.986	Loss: 8.609 Accuracy: 0.411	Loss: 2.028 Accuracy: 0.684
PEDIÁTRICOS: SANOS Y PATOLÓGICOS	"P02", "P04", "P05", "P08", "S11"	"S16"	"P01", "P03", "S17"				738.556		22	Loss: 0.248 Accuracy: 0.984	Loss: 6.514 Accuracy: 0.507	Loss: 1.628 Accuracy: 0.745
ENTRENAMIENTO CON LAS ETIQUETAS: SUJETAR, ABRIR/CERRAR Y 2 MANOS												
CONTROLES Y PEDIÁTRICOS: SANO Y PATOLÓGICO	"P01", "P02", "P03", "S02", "S06", "S07", "S08", "S09", "S10", "S11", "S12", "S13", "S14", "S15", "S16" y "S17".	"S01" y "S03"	"P04", "P05", "P08", "S04" y "S05"	CNN+ "Cono invertido" + GAP	45	5 CNN: 16, 32, 64, 128, 256 + GAP + <i>Dense</i> 128, 64	738.556	DropOut+ Batch Regular. + Regular. L1 y L2	30	Loss: 0.180 Accuracy: 0.973	Loss: 0.320 Accuracy: 0.816	Loss: 0.695 Accuracy: 0.898
							ENTRENAMIENTO AÑADIENDO LA ETIQUETA VERTER					

Tabla 23. Resumen III de los resultados de los experimentos del Bloque II: segmentación de las fases del movimiento *Open/Close a bottle and pour wáter*.

6.2 Discusión

En el presente trabajo fin de grado, se han trabajado dos bloques de experimentos: el reconocimiento de actividades de la vida diaria de los pacientes con parálisis cerebral, y la segmentación de una de estas actividades, en movimientos más simples.

Antes de analizar y evaluar los resultados obtenidos de manera específica para cada bloque de experimentos, vamos a considerar las implicaciones clínicas que puede tener el reconocimiento y segmentación de actividades.

Nuestro sistema automático de reconocimiento de actividades se puede llegar a aplicar en el entorno natural del paciente. Facilitando la rehabilitación desde casa, pudiendo reconocer si se está realizando los movimientos de manera correcta, aumentando la efectividad del tratamiento.

No solo se puede aplicar en entornos naturales, sino también, en el ámbito clínico. Mejorando el reconocimiento de los movimientos por los médicos rehabilitadores, pudiendo ver la afectación de cada parte de la extremidad superior. Como es el caso de (Cao et al., 2019), en su estudio consiguieron desarrollar un sistema capaz de clasificar a los pacientes, según su grado de afectación.

6.2.1 Bloque I: Reconocimiento de actividades de la vida diaria.

En primer lugar, se trata el entrenamiento de una red neuronal para el reconocimiento de diversos movimientos del miembro superior. Entre ellos, *“Reach to grasp and approach”* como *“RTG”*, *“Turn pages”* como *“BOOK”*, *“Put on a hat/helmet”* como *“HAT”* y *“Open/Close a bottle and pour water”* como *“BOTTLE”*.

El primer tema que abordamos es el tamaño de ventana, en este caso, hemos empleado un tamaño de 60 con un solapamiento de 30, al ser movimientos largos, consideramos un tamaño de ventana que engloba 1 segundo.

El entrenamiento de los datos se realizó con una red neuronal convolucional simple, dando lugar a buenos resultados. Sin embargo, vimos que la red confundía la etiqueta de *“BOOK”* con otras etiquetas como: *“HAT”* y *“BOTTLE”*. Esto puede deberse a que dichos movimientos comienzan de manera similar y puede llegar a confundir a la red.

Para los pacientes patológicos vimos que la etiqueta *“BOOK”* era la que más problemas generaba a la hora de ser reconocida por la red. Esto se debe a la dificultad de los niños para realizar esta actividad. Al ser un movimiento combinado de pronosupinación del brazo y de pinza con la mano, los niños no eran capaces de realizar la actividad correctamente.

6.2.2 Bloque II: Segmentación de las fases del movimiento *Open/Close a bottle and pour water*.

Se aborda el tema de entrenamiento de una red neuronal para el reconocimiento de movimientos de miembro superior durante la realización de un movimiento de la vida diaria, como es el movimiento de *Open/Close a bottle and pour water*. A lo largo del estudio se han realizado numerosas pruebas, recopilando información de cómo crear una red y como emplear los datos proporcionados por dispositivos como los sensores inerciales y visión artificial.

El primer tema que vamos a abordar es el tamaño de la ventana junto con su solapamiento. En el experimento 1.1, 1.2 y 1.3 vemos como varían los resultados con los tamaños de ventana de 30, 15, y 45 muestras o *frames*.

El tamaño de 30 *frames* con un solapamiento de 30 no era suficiente para hacer frente al *overfitting*, por lo que se buscó aumentar el número de datos disminuyendo el tamaño de ventana y aumentando el solapamiento.

En este caso el tamaño de ventana de 15 con solapamiento de 10 era demasiado pequeña para que la red fuera capaz de diferenciar algunos movimientos al estar formados por muy pocos *frames* y al haber empleado ventanas de tamaño muy pequeño, da lugar a confusión entre categorías como verter y el resto de las etiquetas.

Por esta razón se decidió aumentar el tamaño de la ventana hasta 45 con un solapamiento de 30. Tratando de encontrar un equilibrio entre un tamaño lo suficientemente pequeño como para que diferencie los movimientos y sea lo suficientemente grande como para que se generen las ventanas necesarias para entrenar correctamente la red.

Los resultados fueron muy positivos comparándolos con los anteriores tamaños de ventanas. Concluimos que los resultados que más nos interesaban los proporcionaban las ventanas de tamaño más grande, en este caso las de 45 con un solapamiento de 30.

La utilización de técnicas de regularización fue fundamental para evitar que el modelo no sufriera *overfitting*. En los entrenamientos con las capas *Dense* se emplearon la técnica de *DropOut*, como podemos ver en el Experimento 1.3, al ver que esta proporcionaba mejores resultados se incluyó en los modelos con redes convolucionales.

En las redes convolucionales además de la técnica de *DropOut* con un 20% de desactivación, se realizaron pruebas usando *Batch Normalization*, como podemos observar en el Experimento 5.1 y con los métodos de regularización L1 y L2, en el Experimento 5.2.

En resumen, llegamos a la conclusión de que emplear los métodos de regularización de *DropOut*, *Batch Normalization*, regularización L1 y regularización L2 eran beneficiosos para el entrenamiento de nuestra red, proporcionando mejores resultados que si no se empleaban.

Se utilizaron redes *Dense* como en el Experimento 3 con una arquitectura de “cono invertido”, sin embargo, se vio en el Experimento 6, que dicha arquitectura, tenía mejor rendimiento en redes Convolucionales.

Además, se empleó una capa de *Global Average Pooling* para reducir la dimensionalidad de las capas, extrayendo las características más relevantes de los datos con un promedio de cada canal de características. Así, conseguimos preservar la información esencial y reduciendo la sensibilidad a pequeñas características de las ventanas, podemos observar dicha mejora en el Experimento 7.

Además, tras la capa de *Global Average Pooling*, favorecía el entrenamiento usar 3 capas *Dense* con 128, 64 y 4 neuronas de salida, como podemos ver en el Experimento 8 y 9.

Por lo que concluimos que la red óptima para clasificar movimientos del miembro superior es una red convolucional con un total de 5 capas de convolución y *Pooling*, de filtros de 16, 32, 64, 128 y 256 con una arquitectura de “cono invertido”. Todas ellas convergen en la capa de *Global Average Pooling*, que la siguen 3 capas *Dense* con 128, 64 y 4 como neuronas de salida, como podemos ver en el Experimento 12.

Otra de las cuestiones que abordamos es la selección de ángulos que participan en el movimiento de *Open/Close a bottle and pour water*. En un primer momento se consideró todos los ángulos pertenecientes a los brazos: "*arm_flex_r*", "*arm_add_r*", "*arm_rot_r*", "*elbow_flex_r*", "*pro_sup_r*", "*arm_flex_l*", "*arm_add_l*", "*arm_rot_l*", "*elbow_flex_l*", "*pro_sup_l*", así como, los ángulos pertenecientes a la espalda: "*lumbar_extension*", "*lumbar_bending*", "*lumbar_rotation*". Se decidió considerar estos últimos ángulos, ya que muchos sujetos inclinaban la espalda a la hora de cogerla botella y el vaso.

No obstante, tras comparar los resultados obtenidos en el Experimento 7 con los del Experimento 8, en los que no se consideraba los ángulos de las lumbares, observamos que al introducir los ángulos de la espalda la red confundía movimientos, entre ellos alejar/acercar y verter. Por lo que se llegó a la conclusión de que era mejor no considerarlos en los entrenamientos.

Las pruebas con los sujetos pediátricos comenzaron seleccionando las etiquetas que se iban a usar. Tras analizar con detalle los vídeos, se llegó a la conclusión de prescindir de la etiqueta de verter, al haber sido descrita como un movimiento que se realizaba agarrando la botella con una mano. Mientras que los niños ejecutaban dicho movimiento sujetando la botella con las dos manos, por lo que se creó una nueva denominada “2Manos”.

Igualmente se eliminó la etiqueta de alejar/acercar, por falta de representatividad.

Además, los niños tenían una mayoritaria afectación en el lado izquierdo, lo que ocasionó que no usaran tanto dicho lado, y los datos de ese lado fueran escasos. Se intentó entrenar una red con los datos del lado izquierdo, sin embargo, fueron insuficientes para llevar a cabo un entrenamiento.

En resumen, tras realizar entrenamientos considerando todos los sujetos pediátricos, tanto sanos como patológicos, en el Experimento 10 y solo los sujetos patológicos en el Experimento 11. Vimos que la red era capaz de reconocer la etiqueta nueva además de las anteriormente utilizadas.

Probamos a realizar un entrenamiento conjunto, los sujetos sanos con los pediátricos sanos y patológicos, con nuestro modelo de red neuronal convolucional, reflejado en el Experimento 13 Y 14.

Concluimos que se pueden realizar entrenamientos de forma conjunta entre los diferentes tipos de sujetos, obteniendo buenas métricas de rendimiento a la hora de clasificar los movimientos que componen el movimiento de *Open/Close a bottle and pour water*. Considerando los movimientos de “Sujetar”, “Abrir/Cerrar”, “Verter” y “2Manos”

6.3 Limitaciones

Una de las principales limitaciones que hemos encontrado ha sido la limitada base de datos y la falta de variabilidad en ella. Dentro de los datos capturados de sujetos sanos, es notable la ausencia de participantes zurdos siendo la gran mayoría de los sujetos diestros. Esto ha ocasionado un desbalanceo de los datos en todos los grupos: *train*, validación y *test*, afectando negativamente a los resultados, provocando un sobre entrenamiento de la red neuronal.

Otra de las limitaciones dentro de la base de datos, ha sido la falta de sujetos pediátricos que cumplieran con las condiciones del estudio, al haber estado en tratamiento de toxina botulínica recientemente, nos ha restringido el número de sujetos patológicos dentro de nuestra base de datos.

Nuevamente, otro inconveniente dentro del estudio ha sido la realización del etiquetado, este ha sido llevado a cabo por un solo experto y de manera manual. En este caso, es un proceso subjetivo donde se puede introducir sesgos y variaciones en los datos etiquetados. Esto puede afectar a la calidad y la imparcialidad de los resultados a la hora de entrenar la red.

Una de las dificultades ha sido el limitado tiempo de estudio, esto ha restringido la cantidad de datos que se ha podido recopilar y analizar. Pudiendo limitar a su vez la robustez de los resultados y la validez de las conclusiones.

El tema de estudio, el análisis de movimientos de tren superior para movimientos de la extremidad superior no consta de una gran investigación previa, como puede ser el análisis de la marcha, que es un tema más estudiado. Lo que ha dificultado el desarrollo de la revisión bibliográfica en la que apoyarse, así como, el enfoque y la línea que seguir durante la investigación.

Este estudio se ha realizado solamente un análisis del movimiento mediante visión artificial. No se ha empleado los sensores inerciales XSens Awinda, debido a la falta de estudios externos, con los que comparar los resultados obtenidos con los sensores.

6.4 Problemas

Durante el desarrollo del TFG es común encontrarse con diversos desafíos y obstáculos que pueden afectar al proceso de investigación y la consecución de los objetivos establecidos.

En primer lugar, el *software* OpenCap al ser una aplicación de código abierto, y acceso gratuito novedosa, ha ocasionado que surgieran diversos fallos dentro de la propia aplicación, dificultando el proceso de aprendizaje de su utilización.

Sin embargo, la respuesta de la propia universidad desarrolladora, la Universidad de Stanford, ha sido positiva a la hora de solucionar los problemas que nos encontrábamos con la mayor brevedad.

Además, durante la utilización de OpenCap en las capturas, surgieron problemas con la sincronización de las dos cámaras, a la hora de iniciar la grabación, ocasionando que los movimientos iniciales no fueran capturados.

El procesamiento de los datos capturados con OpenCap se llevaba a cabo en una nube de cómputo en Amazon Web Services. La opción de montar una instalación en la que procesáramos nuestros datos de forma local se encuentra en desarrollo y de momento no se puede implementar.

Otro de los problemas encontrados es el acceso a los datos de XSens Awinda, siendo de pago, por lo que no se pueden revisar las mediciones hasta que no se paga el acceso al preprocesamiento en la nube. Este proceso tarda cierto tiempo hasta que te dan acceso a la nube.

Finalmente, algunos pacientes con una severa afectación en los miembros superiores no podían llevar a cabo ciertos movimientos como "*Turn pages*" o "*Put on a hat/helmet*".

7. CONCLUSIONES Y LÍNEAS FUTURAS

7.1 Conclusiones

La hipótesis de partida planteada fue la posibilidad de reconocer movimientos relacionados con actividades de la vida diaria en pacientes pediátricos con parálisis cerebral, empleando dispositivos de adquisición.

Tras numerosos experimentos concluimos que, sí es posible reconocer movimientos de miembro superior, empleando redes neuronales de aprendizaje profundo, mediante dispositivos de adquisición como cámaras portátiles.

Además, uno de los objetivos propuestos, la segmentación de las fases de uno de estos movimientos de miembro superior, también fue conseguido. Pudiendo ser posible el reconocimiento de las tareas, por una red neuronal.

Después de realizar una discusión detallada de los resultados obtenidos, procedemos a resaltar las conclusiones obtenidas dentro del estudio de investigación.

En primer lugar, para el reconocimiento de actividades, hemos visto que es posible reconocerlas a través de ventanas de tamaño reducido, como es 1 segundo, empleando redes convolucionales y obteniendo buenos resultados.

En segundo lugar, para la segmentación de miembro superior se ha demostrado la importancia de emplear los ángulos de los brazos, evitando la utilización de los ángulos de las lumbares en los datos obtenidos mediante OpenCap.

En tercer lugar, para un entrenamiento óptimo de las redes neuronales se ha mostrado que las ventanas de tamaño 45 muestras (para datos capturados a 60Hz) proporcionan la suficiente información para englobar los movimientos de interés. Siendo posible el reconocimiento de actividades con ventanas de $\frac{3}{4}$ de segundo.

Por último, se ha validado el uso de redes neuronales convolucionales con una estructura de “cono invertido”, siendo la arquitectura ideal para el reconocimiento de movimientos de miembro superior, usando tanto sujetos sanos adultos como pacientes pediátricos.

7.2 Líneas futuras

Una vez desarrollado el presente estudio, se han descubierto futuras líneas de investigación que pueden complementar y mejorar la cuestión de interés. Entre ellas destacamos las siguientes:

Una de las líneas que se podrían seguir para continuar este estudio sería la ampliación de la base de datos. Aumentar los sujetos participantes, tanto sujetos sanos como patológicos, supondría un impacto positivo de cara a los entrenamientos de la red neuronal. Ya que, el modelo podría generalizar mejor y se

solventaría el problema del *overfitting*. Sobre todo, habría que incluir la presencia de individuos zurdos, dotando de más variabilidad a los conjuntos de entrenamiento, validación y *test*.

Igualmente, otra línea futura podría ser la creación de una base de datos de pacientes sanos pediátricos. Sería interesante realizar un estudio comparativo entre niños sanos y patológicos, de edades similares, realizando los movimientos propuestos para extremidad superior.

Asimismo, otra línea muy relevante sería utilizar sensores inerciales y visión artificial en un estudio comparativo entre pacientes con parálisis cerebral pretratamiento y postratamiento de toxina botulínica. Pudiendo observar la evolución de los miembros afectados y la efectividad del tratamiento. Este caso propuesto se va a llevar a cabo este año al haber sido concedido un proyecto de investigación para realizar este estudio.

Por último, otra continuidad que se podría dar a este proyecto podría ser emplear más dispositivos útiles en el análisis del movimiento, como pudiera ser los sensores de electromiografía, que miden la actividad muscular. Con la finalidad de complementar los resultados y poder comparar entre todos los aparatos empleados la fiabilidad de las mediciones. Además, se podría incluir otros movimientos bimanuales que se realicen en la vida diaria, que se puedan segmentar de forma complementaria al movimiento *Open/Close a bottle and pour water*.

BIBLIOGRAFÍA

- Alliance. (2018). *Manual Ability Classification System (MACS) | Cerebral Palsy Alliance*. <https://cerebralpalsy.org.au/our-research/about-cerebral-palsy/what-is-cerebral-palsy/severity-of-cerebral-palsy/manual-ability-classification-system/>
- Alvear-Puertas, V., Rosero Montalvo, P. D., Peluffo Ordoñez, D. H., & Pijal-Rojas, J. (2017). Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones: Revisión de Literatura. *Enfoque UTE: Facultad de Ciencias de la Ingeniería e Industrias - Universidad UTE*, 8(1), 244-256.
- Argüelles, P. P. (2012). *Parálisis cerebral infantil*.
- Armitano-Lago, C., Willoughby, D., & Kiefer, A. W. (2022). A SWOT Analysis of Portable and Low-Cost Markerless Motion Capture Systems to Assess Lower-Limb Musculoskeletal Kinematics in Sport. *Frontiers in Sports and Active Living*, 3. <https://www.frontiersin.org/articles/10.3389/fspor.2021.809898>
- ASPACE. (2017). *Tipos de parálisis cerebral—ASPACE*. <https://aspacemadrid.org/paralisis-cerebral/tipos-de-paralisis-cerebral/>
- Ávila-Tomás, J. F., Mayer-Pujadas, M. A., & Quesada-Varela, V. J. (2021). La inteligencia artificial y sus aplicaciones en medicina II: Importancia actual y aplicaciones prácticas. *Atención Primaria*, 53(1), 81-88. <https://doi.org/10.1016/j.aprim.2020.04.014>
- Bataineh, M., Marler, T., Abdel-Malek, K., & Arora, J. (2016). Neural network for dynamic human motion prediction. *Expert Systems with Applications*, 48, 26-34. <https://doi.org/10.1016/j.eswa.2015.11.020>
- Benedetto, S. D., Battaglino, A., Abuín-Porras, V., Sánchez-Romero, E. A., Cantero-Tellez, R., Valdes, K., & Villafañe, J. H. (2023). Evaluation of the factors that impact upper limb coordination in children with cerebral palsy: A narrative review. *Retos*, 48, 470-480. <https://doi.org/10.47197/retos.v48.96756>
- Bernad, M. Á. B. (2021, enero 18). *Análisis deportivo mediante el sistema VICON*. Institut Català del Peu. <https://www.institutcataladelpeu.com/analisi-deportivo-mediante-el-sistema-vicon/>
- Borja-Robalino, R., Monleon-Getino, A., & Rodellar, J. (2020). Estandarización de métricas de rendimiento para clasificadores Machine y Deep Learning. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, 30, 184-196.
- Bourke-Taylor, H. (2003). Melbourne Assessment of Unilateral Upper Limb Function: Construct validity and correlation with the Pediatric Evaluation of Disability Inventory. *Developmental Medicine & Child Neurology*, 45(2), 92-96. <https://doi.org/10.1111/j.1469-8749.2003.tb00911.x>
- Cacioppo, M., Marin, A., Rauscent, H., Le Pabic, E., Gaillard, F., Brochard, S., Garlantezec, R., Cretual, A., & Bonan, I. (2020). A new child-friendly 3D bimanual protocol to assess upper limb movement in children with unilateral

- cerebral palsy: Development and validation. *Journal of Electromyography and Kinesiology*, 55, 102481. <https://doi.org/10.1016/j.jelekin.2020.102481>
- Cao, L., Fan, C., Wang, H., & Zhang, G. (2019). A Novel Combination Model of Convolutional Neural Network and Long Short-Term Memory Network for Upper Limb Evaluation Using Kinect-Based System. *IEEE Access*, 7, 145227-145234. <https://doi.org/10.1109/ACCESS.2019.2944652>
- Casares, P. (2015). *Escala AHA (Assisting Hand Assessment) para valoración acciones bimanuales espontáneas*. INEAVA. <https://www.ineava.es/blog/infantil/escala-aha-assisting-hand-assessment-para-valoracion-acciones-bimanuales-espontaneas>
- Castellanos-Ruíz, J., Montealegre-Mesa, L. M., Martínez-Toro, B. D., Gallo-Serna, J. J., & Almanza-Fuentes, O. (2020). Uso de sensores inerciales en fisioterapia: Una aproximación a procesos de evaluación del movimiento humano. *Universidad y Salud*, 23(1), 55-63. <https://doi.org/10.22267/rus.212301.214>
- Choi, R. Y., Coyner, A. S., Kalpathy-Cramer, J., Chiang, M. F., & Campbell, J. P. (2020). Introduction to Machine Learning, Neural Networks, and Deep Learning. *Translational Vision Science & Technology*, 9(2), 14. <https://doi.org/10.1167/tvst.9.2.14>
- COGNEX. (2018). *Introducción a la visión artificial*. https://bcnvision.es/uploads/videotutoriales/uploads/guias%20por%20sectores/introduccion%20a%20la%20vision%20artificial_compressed.pdf
- Edward Joseph Velo Fuentes. (s. f.). *Introducción a los métodos Deep Learning basados en Redes Neuronales*.
- El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? En I. El Naqa, R. Li, & M. J. Murphy (Eds.), *Machine Learning in Radiation Oncology: Theory and Applications* (pp. 3-11). Springer International Publishing. https://doi.org/10.1007/978-3-319-18305-3_1
- Elena Beani et al. (2019). *Actigraph assessment for measuring upper limb activity in unilateral cerebral palsy* | SpringerLink. <https://link.springer.com/article/10.1186/s12984-019-0499-7>
- Elkholy, H. A., Azar, A. T., Magd, A., Marzouk, H., & Ammar, H. H. (2020). Classifying Upper Limb Activities Using Deep Neural Networks. En A.-E. Hassanien, A. T. Azar, T. Gaber, D. Oliva, & F. M. Tolba (Eds.), *Proceedings of the International Conference on Artificial Intelligence and Computer Vision (AICV2020)* (pp. 268-282). Springer International Publishing. https://doi.org/10.1007/978-3-030-44289-7_26
- Ghorbani, S., Mahdavian, K., Thaler, A., Kording, K., Cook, D. J., Blohm, G., & Troje, N. F. (2021). MoVi: A large multi-purpose human motion and video dataset. *PLOS ONE*, 16(6), e0253157. <https://doi.org/10.1371/journal.pone.0253157>
- Ha, S., & Choi, S. (2016). Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors. 2016

- International Joint Conference on Neural Networks (IJCNN)*, 381-388.
<https://doi.org/10.1109/IJCNN.2016.7727224>
- IBM. (2021). *AI in the enterprise*.
- Kumar, R., & Kumar, S. (2023). Effectiveness of Vision Transformers in Human Activity Recognition from Videos. *2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, 593-597.
<https://doi.org/10.1109/InCACCT57535.2023.10141761>
- Lecun, Y., Bengio, Y., & Hinton, G. (s. f.). *Deep learning / Nature*. Recuperado 10 de abril de 2023, de <https://www.nature.com/articles/nature14539>
- Leo, M., Bernava, G. M., Carcagnì, P., & Distante, C. (2022). Video-Based Automatic Baby Motion Analysis for Early Neurological Disorder Diagnosis: State of the Art and Future Directions. *Sensors*, 22(3), Article 3.
<https://doi.org/10.3390/s22030866>
- Li, X., González Navas, C., & Garrido-Castro, J. L. (2017). Fiabilidad y validez de la medida de la movilidad cervical en pacientes con espondiloartritis axial utilizando un sensor inercial. *Rehabilitación*, 51(1), 17-21.
<https://doi.org/10.1016/j.rh.2016.10.002>
- MACS. (2021). *MACS - Manual Ability Classification System*. <https://www.macs.nu/>
- Mahmoud, S. S., Cao, Z., Fu, J., Gu, X., & Fang, Q. (2021). Occupational Therapy Assessment for Upper Limb Rehabilitation: A Multisensor-Based Approach. *Frontiers in Digital Health*, 3.
<https://www.frontiersin.org/articles/10.3389/fdgth.2021.784120>
- Martínez Heras. (2019, noviembre 17). *Precision, Recall, F1, Accuracy en clasificación*. <https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/>
- Mathis, M. W., & Mathis, A. (2020). Deep learning tools for the measurement of animal behavior in neuroscience. *Current Opinion in Neurobiology*, 60, 1-11.
<https://doi.org/10.1016/j.conb.2019.10.008>
- McGrath, J., Neville, J., Stewart, T., & Cronin, J. (2021). Upper body activity classification using an inertial measurement unit in court and field-based sports: A systematic review. *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology*, 235(2), 83-95. <https://doi.org/10.1177/1754337120959754>
- Menor-Rodríguez, M. J., Sevilla Martín, M., Sánchez-García, J. C., Montiel-Troya, M., Cortés-Martín, J., & Rodríguez-Blanque, R. (2021). Role and Effects of Hippotherapy in the Treatment of Children with Cerebral Palsy: A Systematic Review of the Literature. *Journal of Clinical Medicine*, 10(12), 2589.
<https://doi.org/10.3390/jcm10122589>
- Movella. (2021). *Wireless Human Motion tracker for real-time applications*. <https://www.movella.com/products/wearables/xsens-mtw-awinda>
- Movella. (2022, enero 17). *Sensor Placement in Xsens Awinda System*. https://base.xsens.com/s/article/Sensor-Placement-in-Xsens-Awinda-System?language=en_US

- Moya Jiménez, R. C. (2023). *Estudio sistémico de una estructura exoesquelética para la rehabilitación de la atrofia muscular de la mano en personas con artritis reumatoide* [Universitat Politècnica de València]. <https://doi.org/10.4995/Thesis/10251/193057>
- Palermo, M., Lopes, J. M., André, J., Matias, A. C., Cerqueira, J., & Santos, C. P. (2022). A multi-camera and multimodal dataset for posture and gait analysis. *Scientific Data*, 9, 603. <https://doi.org/10.1038/s41597-022-01722-7>
- Park, H., Choi, J. Y., Yi, S., Park, E. S., Shim, D., Choi, T. Y., & Rha, D. (2021). Relationship between the more-affected upper limb function and daily activity performance in children with cerebral palsy: A cross-sectional study. *BMC Pediatrics*, 21, 459. <https://doi.org/10.1186/s12887-021-02927-2>
- Pathirage, C. S. N., Li, J., Li, L., Hao, H., Liu, W., & Ni, P. (2018). Structural damage identification based on autoencoder neural networks and deep learning. *Engineering Structures*, 172, 13-28. <https://doi.org/10.1016/j.engstruct.2018.05.109>
- Randall, M., Imms, C., Carey, L. M., & Pallant, J. F. (2014). Rasch analysis of The Melbourne Assessment of Unilateral Upper Limb Function. *Developmental Medicine & Child Neurology*, 56(7), 665-672. <https://doi.org/10.1111/dmcn.12391>
- Sadowska, M., Sarecka-Hujar, B., & Kopyta, I. (2020). Cerebral Palsy: Current Opinions on Definition, Epidemiology, Risk Factors, Classification and Treatment Options. *Neuropsychiatric Disease and Treatment*, 16, 1505-1518. <https://doi.org/10.2147/NDT.S235165>
- Sanabria S., J. J., & Archila D., J. F. (2011). Detección y análisis de movimiento usando visión artificial. *Scientia et Technica*, 3(49), 180-188.
- Tan, C. C., & Eswaran, C. (2008). Performance Comparison of Three Types of Autoencoder Neural Networks. *2008 Second Asia International Conference on Modelling & Simulation (AMS)*, 213-218. <https://doi.org/10.1109/AMS.2008.105>
- Uhlrich, S. D., & Falisse, A. (s. f.). *Best practices*. Recuperado 3 de abril de 2023, de <https://www.opencap.ai/best-practices>
- Uhlrich, S. D., Falisse, A., Kidziński, Ł., Muccini, J., Ko, M., Chaudhari, A. S., Hicks, J. L., & Delp, S. L. (2022a). *OpenCap: 3D human movement dynamics from smartphone videos* (p. 2022.07.07.499061). bioRxiv. <https://doi.org/10.1101/2022.07.07.499061>
- Uhlrich, S. D., Falisse, A., Kidziński, Ł., Muccini, J., Ko, M., Chaudhari, A. S., Hicks, J. L., & Delp, S. L. (2022b). *OpenCap: 3D human movement dynamics from smartphone videos* [Preprint]. Bioengineering. <https://doi.org/10.1101/2022.07.07.499061>
- Vela, C. C. V., & Ruiz, C. A. V. (2014). Parálisis cerebral infantil: Definición y clasificación a través de la historia. *Revista Mexicana de Ortopedia Pediátrica*, 16(1), 6-10.

- von Marcard, T., Henschel, R., Black, M. J., Rosenhahn, B., & Pons-Moll, G. (2018). Recovering Accurate 3D Human Pose in the Wild Using IMUs and a Moving Camera. En V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer Vision – ECCV 2018* (pp. 614-631). Springer International Publishing. https://doi.org/10.1007/978-3-030-01249-6_37
- Wu, D., Sharma, N., & Blumenstein, M. (2017). Recent advances in vídeo-based human action recognition using deep learning: A review. *2017 International Joint Conference on Neural Networks (IJCNN)*, 2865-2872. <https://doi.org/10.1109/IJCNN.2017.7966210>

