



ULPGC
Universidad de
Las Palmas de
Gran Canaria

Escuela de
Ingeniería Informática



Trabajo de Fin de Grado

Discriminación automática de problemas del desarrollo mediante el análisis de la señal obtenida con un reloj inteligente en la etapa infantil

TITULACIÓN: Grado en Ingeniería Informática

AUTOR: Ana María Jiménez Pérez

TUTORIZADO POR:
Maria Cristina Carmona Duarte

Julio 2024

Agradecimientos

A mi familia por su apoyo incondicional y por siempre creen en mí.

Resumen

Este trabajo se desarrolla con el fin de proporcionar nuevas herramientas informáticas para la ayuda en la detección de problemas en el neurodesarrollo a edad temprana en función de la edad y el género de los niños con el fin de facilitar intervenciones tempranas y adecuadas. Para ello, se va a hacer uso de un reloj inteligente con el que los niños harán una serie de ejercicios mientras se graba el movimiento de estos. Con los datos obtenidos se muestrearán la aceleración y velocidad del movimiento al realizar los ejercicios y a partir de estos datos se discriminará si el niño o niña presenta problemas en el desarrollo neurológico o no.

Palabras clave

TEA (Trastorno del Espectro Autista), TDAH (Trastorno por Déficit de Atención e Hiperactividad), reloj inteligente.

Abstract

This research aims to provide new computer tools to help to detect early neurodevelopmental issues based on the age and gender of children, facilitating timely and appropriate interventions. To achieve this, a smartwatch will be utilized, with which the children will perform a series of exercises while their movements are recorded. The collected data will be used to sample the acceleration and velocity of movement during exercises, from which it will be discerned if the child exhibits neurodevelopmental issues or not.

Key words

ASD (Autism Spectrum Disorder), ADHD (Attention Deficit Hyperactivity Disorder), smartwatch.

Índice general

1. Introducción	1
1.1. Descripción general	2
1.1.1. Organización del documento	3
2. Estado del arte y objetivos iniciales	4
2.1. Estado del arte	4
2.2. Red neuronal convolucional	5
2.3. Objetivos iniciales	6
2.3.1. Aumentar la base de datos con datos de niños entre 0 y 3 años de edad	6
2.3.2. Aportar nuevos algoritmos para obtener parámetros diferenciadores significativos	7
2.3.3. Obtención de diagnóstico	7
3. Competencias específicas y aportaciones del trabajo	8
3.1. Competencias específicas	8
3.2. Aportaciones del trabajo	9
3.3. Aportaciones en el ámbito social	9
4. Metodología implementada	11
4.1. Tecnologías empleadas	13
4.1.1. Python	14
4.1.2. PyCharm	14
4.1.3. Librerías	14
4.1.4. Overleaf	15
4.1.5. Git	15
4.1.6. GitHub	15
4.1.7. Trello	15
4.1.8. TestFlight	15
4.1.9. Anaconda Assistant	15
4.1.10. Jupyter Notebook	16
4.1.11. Chatgpt	16
4.1.12. Equipo de trabajo	16
4.1.13. Smartwatch	16
4.1.14. iPhone	16

5. Desarrollo	17
5.1. Base de datos	17
5.1.1. Estudio de las características de la base de datos previa	17
5.2. Grabación de nuevos datos	18
5.2.1. Resumen del contenido de la base de datos	19
5.3. Desarrollo de diferentes técnicas de parametrización y clasificación adecuadas al número de muestras	21
5.3.1. Preprocesamiento de los datos	21
5.3.2. Algoritmos de parametrización	23
5.3.3. Resultados:	25
5.3.4. Comparación de resultados algoritmos:	33
5.3.5. Aplicación de redes neuronales:	34
5.3.6. Resultados red neuronal:	38
5.3.7. Discusión:	42
5.4. Limitaciones de la investigación:	43
6. Conclusiones y trabajo futuro	44
6.1. Trabajo futuro	44
7. Bibliografía	46
A. Normativa y legislación	50
B. Presupuesto	51

Índice de figuras

4.1. Tablero empleado para la organización y desarrollo del proyecto.	12
5.1. Comparación del muestreo de datos aplicando filtro en la aceleración al lanzar un objeto.	21
5.2. Comparación del muestreo de datos aplicando filtro en la velocidad al lanzar un objeto.	22
5.3. Visualización de los datos de la muestra acotados después de aplicar el filtro de paso bajo.	23
5.4. Nº de picos en niñas en la aceleración.	26
5.5. Nº de picos en niños en la aceleración.	27
5.6. Nº de picos por tiempo de ejecución en niñas en la aceleración.	28
5.7. Nº de picos por tiempo de ejecución en niños en la aceleración.	28
5.8. Tiempo total de ejecución en niñas.	29
5.9. Tiempo total de ejecución en niños.	29
5.10. Tiempo que se tarda en alcanzar la máxima velocidad en niñas.	30
5.11. Tiempo que se tarda en alcanzar la máxima velocidad en niños.	31
5.13. Máxima velocidad alcanzada en niños.	32
5.12. Máxima velocidad alcanzada en niñas.	32
5.14. Antes de normalizar para su uso en el entrenamiento.	35
5.15. Ejemplos de gráficas normalizadas para usarlas en el entrenamiento.	35
5.16. Matriz de confusión para la red 1.	39
5.17. Matriz de confusión para la red 2.	40
5.18. Matriz de confusión para la red 3.	41
5.19. Matriz de confusión para la red 4.	42

Índice de cuadros

4.1. Planificación del proyecto inicial.	13
4.2. Comparación entre la estimación inicial y la duración real de las fases del proyecto	13
5.1. Datos de repetición de ejercicio por edad, género y condición neurológica . .	20
5.2. Datos de repetición de ejercicio por edad, género y condición neurológica reducidos para entrenar correctamente a la red neuronal.	20
5.3. Tabla con los algoritmos empleados y los resultados del P-valor por género y rango de edad	34
5.4. Resultados de las redes neuronales. (C: neurotípico; N: Neurodivergente) . .	38
5.5. Resultados (%) de clasificación para la red neuronal 1	38
5.6. Resultados (%) de clasificación para la red neuronal 2	39
5.7. Resultados (%) de clasificación para la red neuronal 3	40
5.8. Resultados (%) de clasificación para la red neuronal 4	41
B.1. Tabla de precios de productos hardware	51
B.2. Resumen de costos del proyecto	51

Índice de Algoritmos

5.1. Organización de archivos	18
5.2. Número de picos	23
5.3. Número de picos por tiempo de ejecución	24
5.4. Tiempo desde que se empieza hasta que se alcanza la máxima velocidad. . .	25
5.5. Máxima velocidad y aceleración alcanzada	25

Capítulo 1

Introducción

En los últimos años ha habido un aumento de diagnósticos de TEA y TDAH. Esto no solo se debe a que haya aumentado en si la población que lo padece, sino a que también se ha conseguido diagnosticar mejor estos problemas del neurodesarrollo gracias a tener un mayor conocimiento sobre estos trastornos permitiendo a los especialistas hilar más fino a la hora de hacer un diagnóstico. Sin embargo, aun en edades tempranas se observa que existen problemas para discriminar diferentes tipos de trastornos del desarrollo, por ejemplo entre TEA y TDAH. Esto no necesariamente implica un diagnóstico incorrecto, ya que entre el 30 y 40 por ciento de los niños con TEA presentan comorbilidad con otros trastornos. Sin embargo, la falta de un diagnóstico preciso puede impedir que se implementen las intervenciones adecuadas para ayudar al niño a desarrollarse de manera óptima.

Por consiguiente, una detección temprana del TEA es importante para que los niños tengan una enseñanza adecuada donde desarrollen sus habilidades sociales y emocionales. Asimismo, una detección temprana de TDAH puede evitar que el niño presente problemas de regulación emocional, de la conducta y de la atención.

Teniendo en cuenta lo anterior, con este proyecto se busca analizar las diferencias motoras entre niños neurotípicos y neurodivergentes para proporcionar un nuevo método de diagnóstico más preciso.

Para lograr esto, se utilizará un reloj inteligente debido a sus útiles sensores y pantalla táctil, que permitirán registrar el movimiento durante períodos de tiempo determinados. En concreto, los sensores más relevantes para este proyecto son el acelerómetro y el giroscopio. El acelerómetro medirá la aceleración y, a partir de ella, se podrá obtener la velocidad. El giroscopio, por su parte, proporcionará información más completa al indicar la dirección del movimiento.

La finalidad de este proyecto es utilizar los datos recolectados por el reloj inteligente para identificar diferencias motoras significativas entre niños neurotípicos y neurodivergentes, teniendo en cuenta la edad y el género. Con esta información, se pretende mejorar la precisión y fiabilidad de los diagnósticos de TEA y TDAH, facilitando una detección temprana.

Para alcanzar los objetivos de este proyecto, se han aplicado dos métodos. Primero, se

analizaron varios parámetros para identificar cuáles son significativos a la hora de evaluar y poder discernir si un niño tiene problemas del neurodesarrollo o no. Luego, se empleó una red neural para analizar imágenes correspondientes a la aceleración del movimiento en lugar de buscar un parámetro concreto. En específico se ha empleado una red neuronal convolucional (CNN, por sus siglas en inglés). En el apartado 2.2 se detalla más detenidamente su funcionamiento.

1.1. Descripción general

La elección del método de análisis basado en el movimiento se fundamenta en la observación previa de que los niños con TEA y TDAH presentan dificultades en su desarrollo motor [13].

Por un lado, los niños con TEA presentan problemas en sus habilidades visomotoras [30], es decir, en la coordinación entre la vista y el movimiento de manos y otras partes del cuerpo. Por otro lado, los niños con TDAH enfrentan desafíos en la coordinación motora [6], lo que implica dificultades para mover de manera eficiente y sincronizada las diferentes partes del cuerpo.

Dentro del proyecto de investigación en el que se engloba el presente TFT, se realiza la grabación de los siguientes movimientos: caminar, lanzar un objeto, dar un objeto varias veces seguidas y mover de un lado a otro de manera continuada. Este proyecto se centra en el estudio del movimiento de lanzar un objeto debido a su breve duración de 3 segundos y a la facilidad de análisis. Este movimiento es sencillo de identificar, permitiendo eliminar el ruido durante la grabación, gracias a su patrón de velocidad característico: un rápido aumento seguido de una disminución. Este patrón corresponde al levantamiento del brazo para preparar el lanzamiento y la posterior caída del brazo tras soltar el objeto.

El movimiento de lanzar requiere tanto de la coordinación como de las habilidades visomotoras por lo que aportará diferencias entre niños neurotípicos y niños neurodivergentes porque estos últimos harán movimientos menos precisos e inestables. Estas dificultades podrán ser evaluadas con los datos proporcionados por los siguientes sensores del reloj inteligente:

- **Giroscopio:** es un sensor que mide la velocidad angular. Este sensor detecta movimientos rotacionales y cambios en la orientación, proporcionando datos precisos del movimiento y la orientación en tres dimensiones.
- **Acelerómetro:** este sensor mide la aceleración, o la tasa de cambio de la velocidad de un objeto en movimiento. Este sensor puede detectar variaciones en la velocidad de un objeto en varias direcciones.

Con los datos obtenidos a partir de estos sensores, se procede a comparar la señal entre niños del mismo género y edad. El objetivo es entrenar una red neuronal y aplicar algoritmos que identifiquen parámetros en el movimiento que demuestren diferencias significativas entre los niños neurotípicos y los neurodivergentes.

1.1.1. Organización del documento

En este apartado se presenta la estructura del documento, donde se explica brevemente el contenido de cada capítulo y de los apéndices adicionales. Los apartados se organizan de la siguiente manera:

- **Estado del arte y objetivos iniciales:** contextualización del proyecto y definición de los objetivos iniciales.
- **Competencias específicas y aportaciones del trabajo:** definición de las competencias desarrolladas durante el proyecto así como las aportaciones socio-económicas y técnicas aportadas.
- **Metodología implementada:** este apartado define la metodología llevada a cabo para la organización del trabajo.
- **Desarrollo:** explicación de los algoritmos empleados, sus resultados y las tecnologías usadas.
- **Conclusiones y trabajo futuro:** se detalla el resultado del estudio y los siguientes pasos a seguir para mejorar el mismo.
- **Normativa y legislación (Apéndice A):** sección adicional en la que se especifica las leyes tenidas en cuenta durante el desarrollo de este proyecto.
- **Presupuesto (Apéndice B):** sección adicional con la que se analiza cuantitativamente el trabajo realizado.

Capítulo 2

Estado del arte y objetivos iniciales

2.1. Estado del arte

Tradicionalmente, el diagnóstico de TEA y TDAH se ha basado en entrevistas clínicas y cuestionarios estandarizados. Por ejemplo, para el diagnóstico de TEA se utilizan el ADOS (Autism Diagnostic Observation Schedule) [12] y el ADI-R (Autism Diagnostic Interview-Revised) [11]. Estos test aunque son herramientas valiosas para diagnosticar el TEA, su uso exclusivo y los criterios pueden limitar su efectividad, subdiagnosticando ciertos casos y no alineándose completamente con las prácticas clínicas recomendadas [10]. Por otra parte, para el diagnóstico de TDAH, se utiliza el Conners' Rating Scales [7], que se basa en la observación de padres y maestros, lo que puede introducir sesgos debido a percepciones personales y niveles de estrés de los informantes [18]. En resumen, estos métodos tradicionales son subjetivos y los resultados y métodos aplicados pueden variar entre evaluadores.

En los últimos años ha habido un interés creciente en el uso de las tecnologías para el diagnóstico de trastornos del neurodesarrollo. Hay estudios significativos sobre la detección del Trastorno del Espectro Autista (TEA) y el Trastorno por Déficit de Atención e Hiperactividad (TDAH) mencionados a continuación:

- **Estudio 1 - Detección mediante Análisis de Escritura a Mano:** El estudio recopiló muestras de escritura a mano de 30 niños de entre 3 y 5 años, donde se les pidió que dibujaran líneas en zigzag y líneas periódicas. Los investigadores extrajeron diversas características de estas muestras y las utilizaron en diferentes algoritmos de *machine learning* para detectar la coexistencia de TDAH y TEA en los niños capaces de escribir para participar en las pruebas. [29]
- **Estudio 2 - Diagnóstico Temprano con Sensores Multipropósito:** En este estudio, se utilizaron varios sensores para diagnosticar TEA y TDAH en edades tempranas. Los datos obtenidos fueron procesados mediante *machine learning* e inteligencia artificial para mejorar la precisión del diagnóstico y proporcionar un tratamiento más adecuado.[28]

El uso de un reloj inteligente presenta claras ventajas sobre los métodos anteriores. En comparación con el primer estudio, no es necesario que un niño sepa escribir, lo que elimina una barrera significativa para niños más pequeños o con dificultades motoras. Esto amplía el rango de edad y habilidades de los participantes que pueden ser evaluados eficazmente.

Frente al segundo estudio, el uso de un reloj inteligente es considerablemente más cómodo y accesible. Los relojes inteligentes permiten un monitoreo discreto y continuo de la actividad diaria sin la necesidad de múltiples dispositivos. Los datos pueden ser recolectados durante periodos prolongados o limitados por un tiempo previamente establecido, proporcionando una visión más completa y precisa del comportamiento del niño en diferentes contextos.

Además, los relojes inteligentes integran múltiples funciones en un solo dispositivo, lo que reduce la complejidad y los costos asociados con el uso de múltiples sensores. Los datos recopilados pueden ser fácilmente sincronizados con otros dispositivos y sistemas para análisis adicionales, facilitando el acceso y el uso de la información para realizar el estudio.

Este nuevo enfoque demuestra cómo la integración de tecnologías avanzadas puede superar las limitaciones de los métodos tradicionales, mejorando el tiempo necesario para definir un diagnóstico y ofreciendo un nuevo enfoque menos invasivo. Esto proporciona soluciones más inclusivas y eficientes en el campo de la salud infantil.

2.2. Red neuronal convolucional

Una red convolucional se basa en el cortex visual. Es decir, cuando un humano ve una cara, la reconoce porque visualiza unos ojos, una nariz, una boca... y a partir de esos datos construye la imagen e identifica que es una cara. Estas redes tienen un comportamiento similar. Se empieza creando las capas de entrada que se corresponden de forma que si se analiza una imagen de 50x50 píxeles y a color, se usarían 50x50x3 neuronas de entrada.

Una vez se tienen las neuronas de entrada, se empieza a hacer la convolución, que consiste en aplicar varios filtros o "kernel", una matriz generalmente de 3x3, que irá recorriendo todas las neuronas de entrada generando una nueva matriz de salida. Este proceso genera nuevas matrices que destacan características de la imagen original. Si se tiene como entrada una imagen de 50x50x3 y se aplica un filtro de 3x3 con un *padding* adecuado (para mantener el tamaño), el resultado sería una nueva matriz de 50x50 para cada filtro aplicado. Si se utilizan, por ejemplo, 32 filtros, la salida sería una matriz de 50x50x32.

Teniendo en cuenta que se van a aplicar varias capas convolucionales es importante reducir el tamaño de las matrices con el fin de mantener las características más relevantes. Para ello se aplica el "subsampling" o "pooling", como el *max-pooling*, que toma el valor más alto en una región específica. Por ejemplo, con un filtro de 2x2 y un "stride" (número de pasos que se desplaza el "kernel" sobre la matriz) de 2, una matriz de 50x50 se reduciría a una de 25x25.

Este proceso se repetirá varias veces. Con la primera convolución se pueden detectar características simples como líneas o curvas. A medida que se realizan más convoluciones,

se podrán reconocer formas más complejas. Además, entre las capas convolucionales y de "pooling" se aplican funciones de activación como ReLU para introducir no linealidades.

Finalmente, después de varias capas de convolución y "pooling", las redes convolucionales suelen tener capas totalmente conectadas que actúan como clasificadores finales para producir una salida, como la clasificación de la imagen. [3]

Las redes neuronales convolucionales han demostrado ser particularmente eficaces en tareas de reconocimiento y clasificación de imágenes, debido a su capacidad para manejar la variabilidad de las formas en 2D. El término "red neuronal convolucional" indica que la red emplea una operación matemática llamada convolución, que es un tipo especializado de operación lineal. Según [21], las CNN superan a todas las demás técnicas en la tarea de reconocimiento de dígitos manuscritos, lo que justifica su elección en este proyecto para analizar imágenes correspondientes a la aceleración del movimiento.

De forma resumida, estas redes tienen 3 etapas principales [16]:

1. **Convolución:** En esta primera etapa, se aplican varios filtros (*kernels*) en paralelo a la entrada para producir un conjunto de activaciones lineales. Cada filtro se desliza sobre la entrada y realiza una operación de convolución, generando un mapa de características. Esta etapa permite detectar patrones simples en las primeras capas y patrones más complejos en capas más profundas.
2. **Activación no lineal:** Las activaciones lineales resultantes de la etapa de convolución se pasan a través de una función de activación no lineal. Esta operación introduce no linealidades en el modelo, permitiendo que la red aprenda representaciones más complejas de los datos.
3. **Pooling:** En esta etapa, se reduce la dimensionalidad de los mapas de características, disminuyendo la cantidad de parámetros y el costo computacional y ayudando a controlar el sobreajuste.

2.3. Objetivos iniciales

Para el desarrollo de este trabajo, se plantearon como objetivos iniciales aumentar la base de datos con datos de niños entre 0 y 3 años de edad y desarrollar un algoritmo que permita diagnosticar de forma efectiva y rápida problemas en el neurodesarrollo en edades tempranas utilizando un reloj inteligente. A continuación, se define más detalladamente estos objetivos.

2.3.1. Aumentar la base de datos con datos de niños entre 0 y 3 años de edad

Para alcanzar el primer objetivo del proyecto, la autora del mismo solicitó la participación de varias guarderías. A estas instituciones se les facilitó la información sobre el proyecto y, una vez aceptaban participar, se les entregaba el consentimiento informado para obtener la

autorización de los padres para la grabación de los datos. Tras obtener el consentimiento de los padres, se procedió a la grabación de datos con los niños en sus instalaciones.

Sin embargo, la participación de las guarderías y padres no fue la esperada, lo que limitó el aumento de la base de datos en la medida planificada. Adicionalmente, se realizaron pruebas en un hospital, aunque no solo con niños de entre 0 y 3 años. Sin embargo esto permitió obtener más datos para el resto de edades evaluadas en este estudio.

2.3.2. Aportar nuevos algoritmos para obtener parámetros diferenciadores significativos

Para este objetivo se plantearon diferentes algoritmos de análisis para identificar diversos parámetros a partir de las gráficas de velocidad y aceleración. Estos algoritmos se enfocan en medir el tiempo de ejecución del ejercicio, contar el número de picos durante el movimiento, entre otros, para evaluar la consistencia y precisión de los movimientos. La finalidad es determinar qué parámetros son relevantes para identificar si un niño es neurodivergente o no.

2.3.3. Obtención de diagnóstico

Finalmente, este trabajo de investigación busca aportar nuevos métodos para la obtención de diagnósticos, facilitando el proceso tanto para médicos como para niños, y asegurando diagnósticos fiables. Para ello, se usarán los parámetros más significativos y una red neuronal.

Capítulo 3

Competencias específicas y aportaciones del trabajo

3.1. Competencias específicas

Durante el desarrollo de este trabajo, se han puesto en práctica diversas competencias, que se detallan a continuación describiendo cómo estas competencias han sido abordadas y desarrolladas [9].

- **CI6:** *“Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.”*

Durante la elaboración del proyecto, se han desarrollado e implementado varios algoritmos para el procesamiento adecuado de los datos y su posterior análisis, con el objetivo de obtener un algoritmo efectivo para diferenciar entre niños neurodivergentes y neurotípicos.

- **CI7:** *“Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema.”*

Para el manejo de los datos se optó por el uso de listas debido a su versatilidad frente a las tuplas, las cuales tienen el mismo coste computacional. Además, se utilizaron diccionarios para organizar mejor los datos y acceder a ellos fácilmente.

- **CI15:** *“Conocimiento y aplicación de los principios fundamentales y técnicas básicas de los sistemas inteligentes y su aplicación práctica.”*

Para la automatización de diagnósticos, se entrenó una red neuronal. Previamente, se estudió qué tipo de red neuronal era la más adecuada y se investigaron los hiperparámetros óptimos. Estos conocimientos se aplicaron en el código para mejorar el entrenamiento de la red y obtener una mayor fiabilidad en los resultados.

3.2. Aportaciones del trabajo

Con los objetivos definidos en la sección 2.3, se consiguieron varios avances en este estudio definidos a continuación.

En primer lugar, se logró aumentar la base de datos. Se realizaron pruebas con un total de 8 niños en la guardería, aunque este número fue menor al esperado. A pesar de ello, el proceso permitió identificar varias dificultades en la toma de datos. En el apartado 6.1 se detallan estas dificultades y se definen diferentes soluciones que se pueden aplicar en la toma de datos futuras con el fin de obtener unos datos fiables en el futuro.

En segundo lugar, con el análisis de los diferentes parámetros se obtuvieron valores significativos en el número de picos por tiempo de ejecución en la velocidad, en la duración total y en la mayor velocidad y aceleración alcanzadas. Los niños neurotípicos muestran movimientos más fluidos y consistentes, con menos picos en las gráficas. En contraste, los niños neurodivergentes presentan movimientos más erráticos, con más picos y mayor variabilidad en la duración del ejercicio.

En tercer lugar, la aplicación de la red neuronal ofrece un método de identificación más eficiente y rápido. Sin embargo, es crucial contar con un conjunto de datos amplio y variado, ya que, de lo contrario, la red neuronal no será capaz de clasificar correctamente las muestras en sus respectivas categorías. Este modelo de comparación se puede aplicar para el resto de ejercicios que no se han estudiado en este proyecto.

3.3. Aportaciones en el ámbito social

Este trabajo ofrece beneficios directos para la detección y manejo de TEA y TDAH. Además, abre nuevas oportunidades para la aplicación de tecnologías avanzadas en el campo de la salud y contribuye al avance del conocimiento científico en esta área.

Por un lado, la detección temprana de TEA y TDAH en niños tiene un impacto directo en la calidad de vida de los niños y sus familiares. Al facilitar un diagnóstico más rápido y preciso, se pueden implementar intervenciones tempranas que mejoren el desarrollo y las oportunidades de los niños afectados, reduciendo así los costos a largo plazo asociados con tratamientos tardíos y apoyo educativo adicional.

Por otro lado, este trabajo contribuye al avance de las tecnologías de la salud, demostrando la aplicación práctica de dispositivos tecnológicos para ayudar en la evaluación de trastornos neuropsiquiátricos.

Desde una perspectiva científica, este estudio aporta nuevos conocimientos sobre los patrones de movimiento asociados en niños neurodivergentes. La investigación realizada amplía la comprensión de cómo se pueden utilizar los datos de sensores para diferenciar entre niños neurotípicos y neurodivergentes. Los resultados obtenidos pueden servir como base para futuras investigaciones y para el desarrollo de modelos más sofisticados de diagnóstico basados

*CAPÍTULO 3. COMPETENCIAS ESPECÍFICAS Y APORTACIONES DEL TRABAJO*¹⁰

en datos.

Capítulo 4

Metodología implementada

Para el desarrollo de este proyecto se ha aplicado la metodología ágil *Kanban*, implementada mediante tableros. Esta metodología permite una visualización clara del flujo de trabajo, proporcionando un mayor control sobre el estado del proyecto. Además, facilita la limitación del trabajo en proceso, lo que ayuda a centrarse en objetivos específicos sin causar retrasos. Para el desarrollo de este proyecto se puso como máximo que hubieran dos trabajos en progreso debido a que fue desarrollado por una persona.

Kanban es flexible en cuanto a los estados que se pueden asignar en el tablero. Para la realización de este trabajo, en un principio se propuso añadir tres columnas adicionales ("Análisis", "Pruebas/Revisión" y "Documentación") a parte de las tres columnas principales: "Por Hacer", "En Desarrollo" y "Finalizado". Sin embargo, al final se optó por usar únicamente las tres columnas principales debido a que se acumulaba mucho trabajo que estaba en proceso ya que las columnas que se querían añadir de "Análisis" y "Pruebas/Revisión" realmente corresponden al trabajo en proceso.

Para el uso de esta metodología, se utilizó *Trello* [5], una herramienta en línea que permite crear tableros para realizar el seguimiento del trabajo. Se creó un tablero con las tres columnas especificadas anteriormente y varias tarjetas que representan las tareas a desarrollar, identificadas con etiquetas para indicar a cuál de las cuatro fases pertenece cada una 4.1.

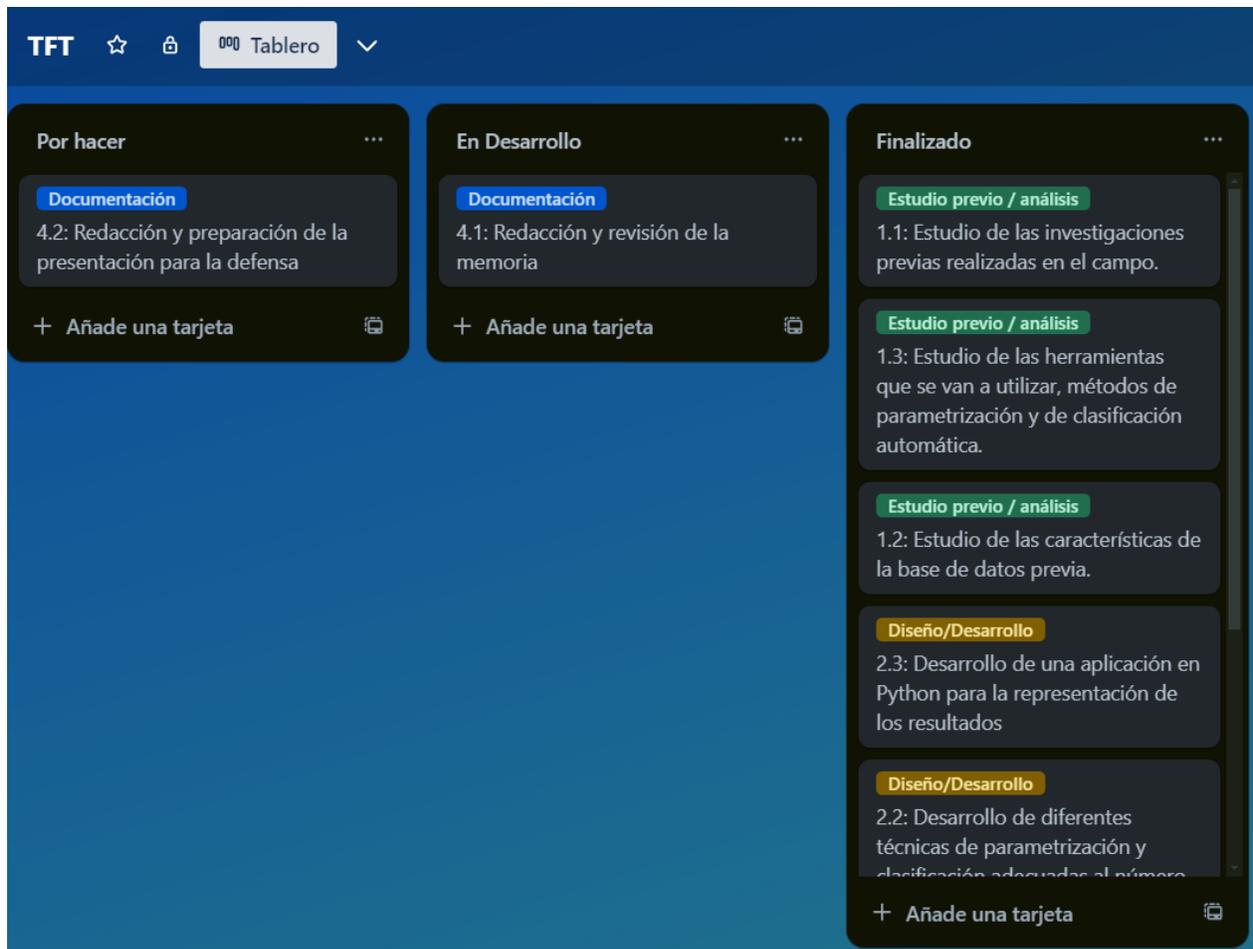


Ilustración 4.1: Tablero empleado para la organización y desarrollo del proyecto.

Para el desarrollo del proyecto se han llevado a cabo un total de 10 tareas, divididas en 4 fases. En un principio se estimó que la duración de las fases sería la que se muestra en la tabla 4.1.

Sin embargo, al final se dedicó menos tiempo a las fases de Estudio previo y Documentación. Esto se debió a que se proporcionó mucha documentación, la cual, en general, fue fácil de leer porque la información relevante se encontraba de manera sencilla. Por otro lado, el tiempo ahorrado en estas fases se destinó a las fases de Diseño y Evaluación, aumentando así el tiempo dedicado a estas respecto a lo estimado inicialmente. La distribución de horas para cada fase es la que se muestra en la tabla. 4.2.

Cuadro 4.1: Planificación del proyecto inicial.

Fases	Duración Estimada (horas)	Tareas
Estudio previo / Análisis	60	Tarea 1.1: Estudio de las investigaciones previas realizadas en el campo.
		Tarea 1.2: Estudio de las características de la base de datos previa.
		Tarea 1.3: Estudio de las herramientas que se van a utilizar, métodos de parametrización y de clasificación automática.
Diseño/Desarrollo/Implementación	100	Tarea 2.1: Grabación de nuevos datos
		Tarea 2.2: Desarrollo de diferentes técnicas de parametrización y clasificación adecuadas al número de muestras.
		Tarea 2.3: Desarrollo de una aplicación en Python para la representación de los resultados.
Evaluación/Validación/Prueba	80	Tarea 3.1: Evaluación de los resultados.
		Tarea 3.2: Mejora de los algoritmos de detección y parametrización.
Documentación / Presentación	60	Tarea 4.1: Redacción y revisión de la memoria.
		Tarea 4.2: Redacción y preparación de la presentación para la defensa.

Cuadro 4.2: Comparación entre la estimación inicial y la duración real de las fases del proyecto

Fase	Estimación Inicial (horas)	Duración Real (horas)
Estudio previo / Análisis	60	40
Diseño/Desarrollo/Implementación	100	120
Evaluación/Validación/Prueba	80	90
Documentación / Presentación	60	50

4.1. Tecnologías empleadas

En este apartado, se detallan las tecnologías utilizadas para llevar a cabo este proyecto así como las librerías de *Python* empleadas. Se presentarán tanto las herramientas de software como las herramientas de hardware que fueron necesarias para cumplir con los objetivos planteados.

4.1.1. Python

Python es un lenguaje de programación de alto nivel, interpretado, dinámico y multiplataforma. Es de código abierto, administrado por *Python Software Foundation* (PSF) [26]. Es ampliamente utilizado en diversos campos, incluyendo la ciencia de datos y el *machine learning*, lo cual lo hizo atractivo para este proyecto basado en la automatización de diagnósticos.

4.1.2. PyCharm

PyCharm es un entorno de desarrollo integrado (IDE) elaborado por *JetBrains* específicamente para el lenguaje de programación *Python* [19]. Proporciona herramientas para la escritura, edición y depuración de código *Python*, así como soporte para varios *frameworks* y bibliotecas.

4.1.3. Librerías

- **numpy:** Librería especializada en el cálculo numérico y el análisis de datos. Empleada en el desarrollo del proyecto por proporcionar varios métodos para poder llevar a cabo el análisis de los datos.
- **scipy:** Librería científica de Python de la que se han usado los siguientes módulos:
 - **scipy.signal:** Submódulo para el procesamiento de señales. Específicamente se han usado los métodos *"butter"* y *"lfilter"* para realizar el filtrado.
 - **scipy.stats:** Submódulo para análisis estadístico. En concreto se ha usado para realizar el análisis ANOVA.
 - **scipy.integrate:** Submódulo para realizar integración numérica de funciones. Se ha usado para calcular la integral acumulada de los datos de aceleración en función del tiempo para obtener la velocidad.
- **os:** Módulo que permite interactuar con el sistema operativo. Se ha usado para acceder a los datos.
- **shutil:** Módulo que proporciona varias funciones para la manipulación de archivos y directorios. Se ha usado para organizar la base de datos.
- **matplotlib:** Librería para la creación de gráficos. Concretamente se ha usado el submódulo "matplotlib.pyplot".
- **pandas:** Librería especializada en la manipulación y análisis de datos estructurados. En este proyecto, se ha utilizado para crear *DataFrames* que faciliten el análisis estadístico.

4.1.4. Overleaf

Overleaf es un editor de documentos *LaTeX* en línea que se ha utilizado para la redacción de la memoria del proyecto [8]. El principal motivo de su elección son las facilidades que ofrece para estructurar cómodamente el documento y para realizar de manera sencilla y ordenada las referencias bibliográficas.

4.1.5. Git

Git es un sistema de control de versiones [27] que ha sido utilizado para llevar un control del código realizado en el proyecto. Este sistema permite gestionar de forma fácil y eficiente los cambios realizados en el código, asegurando un trabajo seguro al tener la opción de revertir cambios y comparar diferentes versiones del código.

4.1.6. GitHub

GitHub es un servicio de alojamiento de repositorios basado en la nube que ha sido utilizado para almacenar y gestionar el código fuente del proyecto [14]. Actúa como un repositorio central donde los desarrolladores pueden colaborar, compartir y revisar código de manera eficiente. Además de ofrecer control de versiones mediante *Git*, *GitHub* proporciona herramientas adicionales facilitando así un flujo de trabajo ágil.

4.1.7. Trello

Trello [5] es una herramienta para la gestión de proyectos que ha sido utilizada para organizar y visualizar el progreso del proyecto. Permite crear tableros dinámicos donde se pueden agregar listas y tarjetas, ofreciendo una herramienta visual para ver el estado del proyecto en tiempo real. Cada tarjeta se puede enriquecer con descripciones, fechas de vencimiento, etiquetas, comentarios y datos adjuntos.

4.1.8. TestFlight

TestFlight [2] es una plataforma de *Apple* que permite a los desarrolladores distribuir versiones beta de sus aplicaciones. Se ha utilizado para poder ejecutar la aplicación con la que se llevaba a cabo la toma de datos.

4.1.9. Anaconda Assistant

Anaconda es una plataforma de software que facilita la instalación y gestión de entornos de programación en *Python* y *R* [1]. En este proyecto, se utilizó *Anaconda* para instalar y

gestionar las bibliotecas necesarias, como *Keras* y *TensorFlow*, ya que no era posible instalarlas localmente. Inicialmente se consideró usar *Google Colab*, pero debido a las limitaciones de GPU, resultó inviable para completar el proyecto de manera efectiva.

4.1.10. Jupyter Notebook

Jupyter Notebook [31] es una aplicación web que permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. Utilizando *Anaconda*, se empleó *Jupyter Notebook* para desarrollar y probar la red neuronal, aprovechando la facilidad que ofrece para combinar código y visualizaciones en un solo documento, así como para utilizar las bibliotecas necesarias.

4.1.11. Chatgpt

ChatGPT [17] es una herramienta de inteligencia artificial. Se ha usado para mejorar la redacción del documento, optimizar el código y hacer que sea más limpio y eficiente.

4.1.12. Equipo de trabajo

El desarrollo de este proyecto se ha realizado en un ordenador portátil *OMEN* proporcionado por *HP* [25]. Este equipo personal cuenta con la mayoría del software mencionado en el apartado anterior y el acceso a los recursos en línea siendo el equipo principal de trabajo.

4.1.13. Smartwatch

Para la toma de datos, se ha utilizado un *Apple Watch Serie 7 (45mm)* [24] debido a sus sensores incorporados. En él se ejecutaba el software *WData*, con el fin de grabar el movimiento realizado por los niños.

4.1.14. iPhone

Por último se especifica el sistema con el que se ejecutaba la aplicación *TestFlight* ya descrita y el cual también ha sido usado como intermediario entre el reloj y la base de datos, el cual ha sido un *iPhone SE* [23].

Capítulo 5

Desarrollo

5.1. Base de datos

5.1.1. Estudio de las características de la base de datos previa

Inicialmente, la base de datos estaba organizada de la siguiente manera: existía un archivo principal que contenía el diagnóstico de cada niño. Dentro de este archivo principal, se encontraba un directorio para cada niño identificado por un ID único, seguido por su género y edad, todo separado por un guión bajo (ejemplo: "ID_edad_género"). Dentro de cada uno de estos directorios, se almacenaban los archivos correspondientes a las pruebas realizadas a cada niño.

Junto con la base de datos, se adjuntó un *Excel* que especificaba el identificador de cada niño y qué pruebas había realizado. Se comprobó si en la base de datos se encontraban todas las pruebas especificadas en el *Excel* y se detectaron varias deficiencias en la base de datos, entre las cuales destacaba la ausencia de muchos archivos en los directorios correspondientes de cada niño, es decir, faltaban las grabaciones de las pruebas. Para solucionar este problema, se procedió a la incorporación manual de los archivos, asegurándose de que cada archivo estuviera correctamente definido y ubicado en su directorio correspondiente.

Una vez que la base de datos estuvo completa y verificada, se reorganizó toda la información en un nuevo directorio estructurado de la siguiente manera: primero se separaron los datos en dos categorías, "Neurotípico" y "Neurodivergente". Dentro de cada una de estas categorías, los datos se organizaron por género (masculino y femenino) y, posteriormente, por edad.

Debido a los problemas encontrados durante la revisión de la base de datos, se propuso una nueva nomenclatura especificada en el apartado 5.2 para la definición de los archivos, facilitando así la organización de los datos.

Con este nuevo método de nomenclatura se desarrolló un programa para que se organicen de forma automática los datos en un archivo por cada niño. Lo primero es comprobar si

el niño es neurotípico o neurodivergente. Si es neurotípico, se guardará bajo el directorio 0000_H1, donde H1 corresponde al sitio en el que se hayan realizado las pruebas; y si no, se guardará en el directorio 1000_H1. Luego, se creaba la carpeta específica para cada niño con la nomenclatura: "ID_edad_genero".

Una vez se encuentran bien agrupados los datos según si tienen problemas en el neurodesarrollo o no, se procede a mover los directorios de cada niño a un directorio general con la estructura: "/Neurotipico/género/edad" cogiendo los datos del directorio que identifica a cada niño "ID_edad_genero" para moverlo al directorio correcto.

```

1  for files in os.listdir(directorio_path):
2      partes = files.split('_')
3      edad = partes[1]
4      sexo = partes[2]
5
6      estado_neurodesarrollo = 'Neurotipico' if flagSano else
7          'Neurodivergente'
8      destino_sexo = 'male' if sexo == 'M' else 'female'
9      destino_final = os.path.join(destino, estado_neurodesarrollo,
10         destino_sexo, edad)
11
12     if not os.path.exists(destino_final):
13         os.makedirs(destino_final)
14     shutil.move(os.path.join(directorio_path, files), destino_final)

```

Algoritmo 5.1: Organización de archivos

5.2. Grabación de nuevos datos

La grabación de nuevos datos se ha realizado utilizando un reloj inteligente con una aplicación que permite establecer tanto el tiempo de grabación como el nombre del archivo. Para facilitar el agrupamiento de los datos según la condición de neurodesarrollo, el género y la edad, se ha empleado una nueva nomenclatura para los nombres de los archivos, siguiendo el formato: "sitio_bitNeurodesarrollo_edad_bitGénero_identificador_prueba_" donde:

- **sitio:** Lugar donde se realizaron las pruebas.
- **bitNeurodesarrollo:** Indica si el niño es neurotípico (0) o no (1).
- **edad:** Se especificó la edad en meses.
- **bitGénero:** Género del niño, donde 0 representa masculino y 1 representa femenino.
- **identificador:** Número identificador único asociado a cada niño.
- **prueba:** Tipo de prueba realizada.

La toma de datos consistía en ponerle el reloj a un niño de entre 0 y 3 años y pedirle que realizara ciertos ejercicios usando la mano en la que tenía el reloj. Los ejercicios realizados son los siguientes:

- Andar por 1 minuto.
- Lanzar un objeto 10 veces en repeticiones de 3 segundos.
- Mover un objeto de izquierda a derecha varias veces repitiendo el ejercicio 3 veces en periodos de 20 segundos.
- Dar un objeto y cogerlo varias veces repitiendo el ejercicio 3 veces en periodos de 20 segundos.

Durante el desarrollo de estas pruebas con los niños pequeños hubieron algunas dificultades, ya que no siempre seguían las instrucciones adecuadamente, por lo que algunas grabaciones se hacían con un movimiento erróneo.

Esto supone un problema a la hora de muestrear los datos y cortar el movimiento porque es posible que se esté cogiendo un movimiento no válido, por ejemplo que en vez de lanzar simplemente suelte el objeto y lo deje caer o que empiece a lanzar muy tarde y solo se grabe la mitad del movimiento.

Esto condiciona la toma de datos aunque siempre es posible eliminar los archivos que se sepan incorrectos tras tener los datos en la base de datos. Sin embargo, esto hace que no sea un proceso automático.

Más adelante en el apartado 5.4, se especifican las limitaciones en la investigación y la solución estas limitaciones para hacer más fácil la toma de datos y así evitar errores.

5.2.1. Resumen del contenido de la base de datos

En esta sección se va a especificar la cantidad de datos disponibles para realizar las pruebas, que corresponden a los datos de lanzar un objeto. Para obtener el número de muestras que se van a aplicar se tiene en cuenta que la aceleración alcanzada sea de al menos 5 m/s^2 ya que si no denota que no se realizó el movimiento. Si un niño no tiene ninguna muestra que valga, no se cuenta como sujeto de prueba por lo que no se añade en la tabla de resumen.

Estos fueron usados en su completitud para realizar la búsqueda de parámetros significativos que diferencien ambos casos de estudio 5.2. Por otra parte, para la red neuronal se prescindió de varios datos con el objetivo de que para cada niño hubiese el mismo número de gráficas y estudiar solo los casos con bastantes datos. Esta reducción de los datos se puede apreciar en la tabla

Etiqueta	Edad	Sujetos	Total Repeticiones
Niña Neurodivergente	0-3	0	0
	4-7	9	61
	8-11	12	75
	12-15	7	22
	16-19	2	10
Niña Neurotípica	0-3	3	6
	4-7	3	29
	8-11	25	198
	12-15	13	111
	16-19	8	73
Niño Neurodivergente	0-3	1	3
	4-7	29	142
	8-11	18	104
	12-15	19	99
	16-19	0	0
Niño Neurotípico	0-3	2	8
	4-7	7	60
	8-11	36	294
	12-15	21	174
	16-19	5	46

Cuadro 5.1: Datos de repetición de ejercicio por edad, género y condición neurológica

Etiqueta	Edad	Sujetos	Total Repeticiones
Niña Neurodivergente	8-11	12	34
	12-15	7	21
Niña Neurotípica	8-11	25	75
	12-15	13	39
Niño Neurodivergente	8-11	18	50
	12-15	19	51
Niño Neurotípico	8-11	36	106
	12-15	21	60

Cuadro 5.2: Datos de repetición de ejercicio por edad, género y condición neurológica reducidos para entrenar correctamente a la red neuronal.

5.3. Desarrollo de diferentes técnicas de parametrización y clasificación adecuadas al número de muestras

En este apartado se definirán los algoritmos empleados sobre los datos así como pasos previos realizados con el objetivo de obtener mejores resultados.

5.3.1. Preprocesamiento de los datos

Desde un principio se propuso obtener la velocidad y aceleración de los datos muestreados. Una vez se obtenían, se aplicó un filtro con el objetivo de obtener una señal con menos ruido. Posteriormente se desarrolló un algoritmo de corte para poder acotar la muestra al momento en el que se llevo a cabo el ejercicio.

El filtro aplicado es el filtro de paso bajo diseñado para atenuar las frecuencias altas y eliminar el ruido en la señal, teniendo en cuenta la frecuencia de *Nyquist*. Este filtro necesita que se especifiquen dos parámetros: la frecuencia de muestreo, establecida en 100 Hz, y la frecuencia de corte, que se establece mediante un valor normalizado. En el código, la frecuencia de corte normalizada es 0.8, lo que significa que la frecuencia de corte real es $0.8 \times 50\text{Hz} = 40\text{ Hz}$ (dado que la frecuencia de *Nyquist* es la mitad de la frecuencia de muestreo, es decir, 50 Hz).

Por lo tanto, este filtro permitirá el paso de componentes de frecuencia por debajo de 40 Hz y atenuará aquellas que se encuentren por encima de esta. Dado que la frecuencia de corte de 40 Hz está por debajo de la frecuencia de *Nyquist*, se asegura una adecuada filtración sin distorsionar las componentes críticas de la señal 5.1, 5.2.

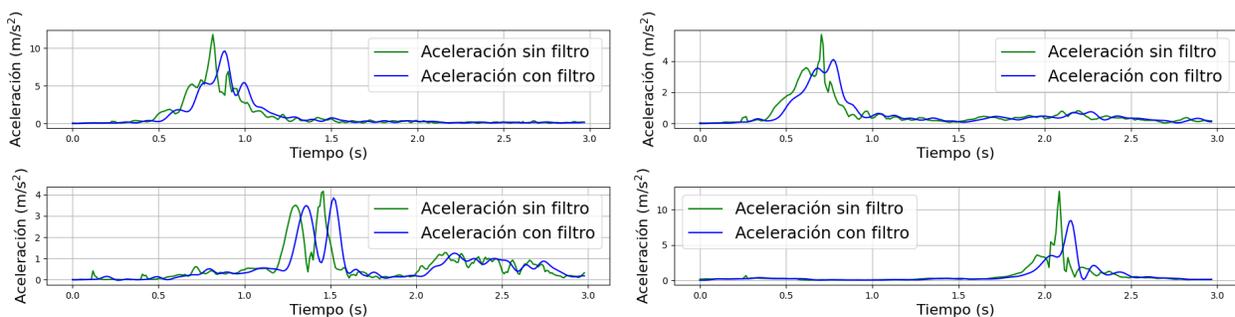


Ilustración 5.1: Comparación del muestreo de datos aplicando filtro en la aceleración al lanzar un objeto.

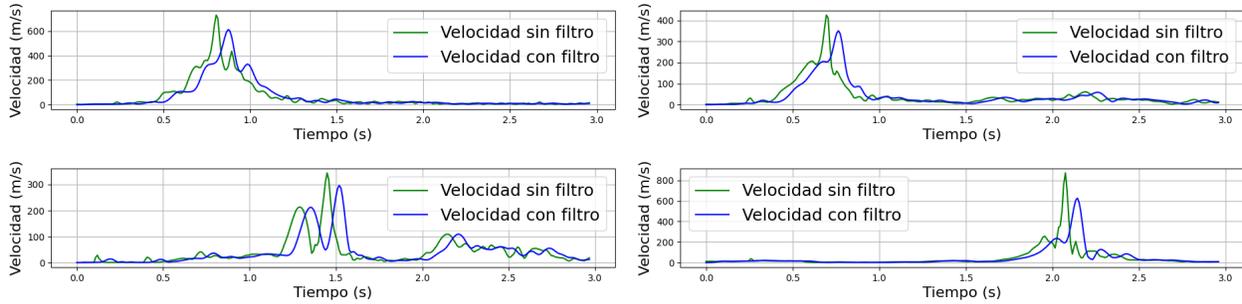


Ilustración 5.2: Comparación del muestreo de datos aplicando filtro en la velocidad al lanzar un objeto.

Una vez que se logró obtener una señal más clara, se procedió a extraer únicamente la parte relevante de la grabación correspondiente al movimiento, basándose en la aceleración obtenida. Para ello, se implementaron y evaluaron dos algoritmos destinados a establecer los puntos de corte en la gráfica de la señal:

1. **Algoritmo de búsqueda del pico más bajo y el más alto:** El primer algoritmo se centró en identificar el pico más bajo, bajo la premisa de que el niño tendría el brazo en reposo, seguido del pico más alto, correspondiente al momento en que el niño alcanzaba la mayor velocidad y aceleración durante el movimiento.

Aunque inicialmente este algoritmo parecía funcionar correctamente, se presentaron varios inconvenientes. Algunos niños comenzaban a realizar el movimiento antes de que se iniciara la grabación, y en ocasiones, el pico más alto se encontraba casi al inicio del muestreo de los datos, lo que dificultaba la correcta segmentación del movimiento.

2. **Algoritmo de búsqueda del pico más alto y picos mínimos adyacentes** Para abordar los problemas identificados con el primer algoritmo, se desarrolló un segundo algoritmo. Este nuevo enfoque consistió en identificar primero el pico más alto y, a continuación, buscar los picos más bajos antes y después de este. Este método permitió delimitar de manera más precisa la parte del movimiento relevante. En la imagen 5.3 se puede observar que se descarta gran parte de la grabación y se selecciona únicamente la parte donde se realiza el movimiento. En esta sección se aprecia claramente la aceleración al subir el brazo para lanzar y la disminución de esta al tirar el objeto.

La implementación de este segundo algoritmo resultó en una segmentación más adecuada de la señal de movimiento, superando las limitaciones encontradas con el primer enfoque. Esta mejora garantiza una evaluación más precisa y consistente del movimiento infantil registrado.

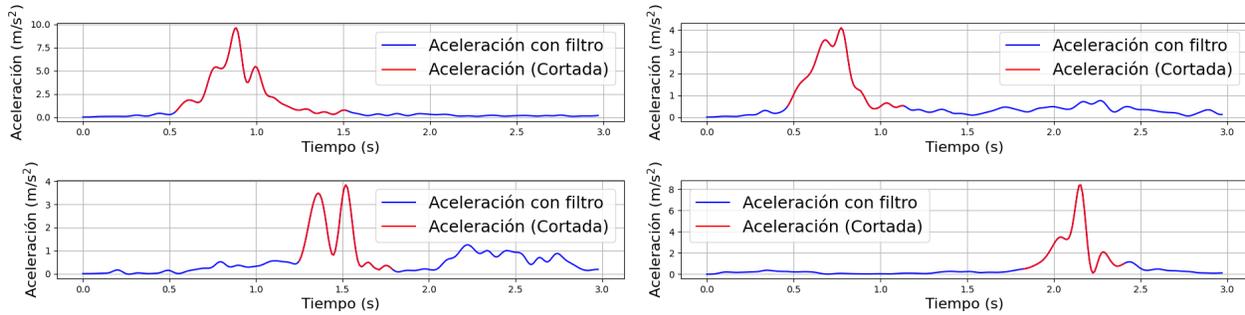


Ilustración 5.3: Visualización de los datos de la muestra acotados después de aplicar el filtro de paso bajo.

Tras aplicar el algoritmo de corte a los datos y retirar el filtro, se realizaron las representaciones gráficas de la velocidad y la aceleración sobre las que se aplicaron diferentes algoritmos de parametrización.

5.3.2. Algoritmos de parametrización

Con el objetivo de encontrar el parámetro más relevante en la discriminación entre ambos casos de estudio, se aplicaron diferentes algoritmos sobre la parte acotada de los datos sin aplicar el filtro. Para realizar las comparaciones, se ha separado por género y por rango de edades.

A continuación, se describen los algoritmos desarrollados:

- **Número de picos:**

Se cuenta el número de picos presentes en la aceleración y en la velocidad. Estos picos representan la variabilidad en los movimientos, indicando si la aceleración y la velocidad aplicadas son constantes. Movimientos más erráticos y menos fluidos son indicadores de las dificultades en la coordinación y el control motor que los niños neurodivergentes pueden experimentar.

```

1 def find_peaks(self, data):
2     peaks = (np.roll(data, 1) < data) & (np.roll(data, -1) < data)
3     return peaks.sum()

```

Algoritmo 5.2: Número de picos

Para ello, se aplica una función anterior, que toma como parámetro un *array*, correspondiente a los datos de aceleración o velocidad según aplique. La función identifica los elementos del *array* que son mayores que el elemento anterior y los elementos que son mayores que el siguiente. Al combinar ambas comparaciones, se pueden encontrar los picos, es decir, los elementos que son mayores que sus vecinos inmediatos.

- **Número de picos por tiempo de ejecución:**

Se cuenta el número de picos que hay en la aceleración y en la velocidad durante el tiempo de ejecución. Esto se llevó a cabo porque si hay varios picos en un corto periodo de tiempo indica poca fluidez y control del movimiento. Haciendo uso del método desarrollado anteriormente, se calcula el tiempo de ejecución tomando como inicio y fin la zona acotada. Luego, se divide el número de picos por el tiempo de ejecución para obtener la tasa de picos por unidad de tiempo.

La parte más relevante del código usado se muestra a continuación:

```

1 def get_peak_counts_per_second(self, files):
2     # ...
3     if measure == "Velocity":
4         num_peaks = self.analyzer.find_peaks(Vt[start:end])
5     elif measure == "Acceleration":
6         num_peaks = self.analyzer.find_peaks(A[start:end])
7     else:
8         raise ValueError("Medida no valida. Debe ser
9             'Acceleration' o 'Velocity'.")
10
11     duration = t[end] - t[start]
12     peaks_per_second = num_peaks / duration
13     # ...

```

Algoritmo 5.3: Número de picos por tiempo de ejecución

- **Tiempo total en realizar el ejercicio:**

Se mide el tiempo total del ejercicio el cual corresponde a la zona acotada, donde se ha eliminado el ruido utilizando los datos de la aceleración. En este caso, la duración es la misma tanto si se emplean datos de aceleración como de velocidad, ya que el intervalo de tiempo considerado es el mismo.

Medir solo el tiempo total del ejercicio permitirá evaluar la rapidez y eficiencia del movimiento con lo que se puede detectar anomalías en la ejecución, como la falta de coordinación, haciendo que se tarde más en completar el ejercicio.

- **Tiempo desde que se empieza hasta que se alcanza la máxima velocidad:**

Se mide el tiempo que hay desde que el niño empieza a realizar el ejercicio hasta que alcanza su máxima velocidad o aceleración, que a efectos prácticos es lo mismo. Para ello, se busca el índice donde se encuentra el pico máximo y luego se le resta el tiempo de inicio.

Este intervalo de tiempo permite evaluar la rapidez con la que el niño puede generar un movimiento máximo, proporcionando información sobre su capacidad de aceleración y la eficiencia de su sistema neuromotor.

A continuación se puede ver una muestra del código implementado:

```

1 def get_peak_counts_per_second(self, files):
2     # ...
3     if measure == "Velocity":
4         indice_maximo = np.where(Vt[start:end] ==
5                                 Vt[start:end].max())[0][0]
6     elif measure == "Acceleration":
7         indice_maximo = np.where(A[start:end] ==
8                                 A[start:end].max())[0][0]
9     else:
10        raise ValueError("Medida no valida. Debe ser 'Acceleration' o
11                          'Velocity'.")
12
13    time_to_peak = np.abs(t[indice_maximo] - t[start])
14    # ...

```

Algoritmo 5.4: Tiempo desde que se empieza hasta que se alcanza la máxima velocidad.

- **Valor de la máxima velocidad y aceleración alcanzada:**

Se obtiene el valor de la máxima aceleración y velocidad que alcanza el niño aplicando la función también empleada en el algoritmo anterior.

Por un lado, la máxima aceleración indicará la capacidad del niño para generar fuerza rápidamente, mientras que la máxima velocidad indica la eficiencia y capacidad máxima de su movimiento.

```

1 def find_peaks(self, data):
2     peaks = (np.roll(data, 1) < data) & (np.roll(data, -1) < data)
3     return peaks.sum()

```

Algoritmo 5.5: Máxima velocidad y aceleración alcanzada

5.3.3. Resultados:

En esta sección se comentan los resultados de cada algoritmo de forma individual para evaluar su viabilidad. Para ello, se analizan los *boxplots* con los resultados obtenidos. Los *boxplots* se diferencian entre dos grupos: neurotípicos y neurodivergentes, para ambos géneros (masculino y femenino). Cada grupo está representado por un *boxplot* que permite visualizar y comparar la distribución de los datos.

Como se han realizado pruebas con la velocidad y la aceleración para cada algoritmo, en cada caso se mostrará solo aquella que haya dado mejores resultados para ambos géneros.

- **Número de picos:**

Como se puede observar en las figuras 5.4 y 5.5, existe una tendencia general en la que los niños y niñas neurodivergentes presentan un mayor número de picos en comparación con los neurotípicos. Esto sugiere que los movimientos de los neurodivergentes son más erráticos y menos fluidos, posiblemente debido a problemas de coordinación motora.

En las niñas, se observa una diferencia notable en las edades de 4 a 7 y 8 a 11 años, donde la mediana y la media del número de picos son mayores para las neurodivergentes.

En los niños, esta tendencia es notable en los grupos de edad de 4 a 7 y 8 a 11 años. En estos grupos, los niños neurodivergentes muestran una mayor cantidad de picos, aunque la variabilidad dentro de cada grupo es mayor, con numerosos valores atípicos.

Sin embargo, también se observa una considerable superposición para los dos géneros en los datos de ambos grupos, lo que indica que, aunque hay una tendencia, no es lo suficientemente clara en todos los casos para ser concluyente.

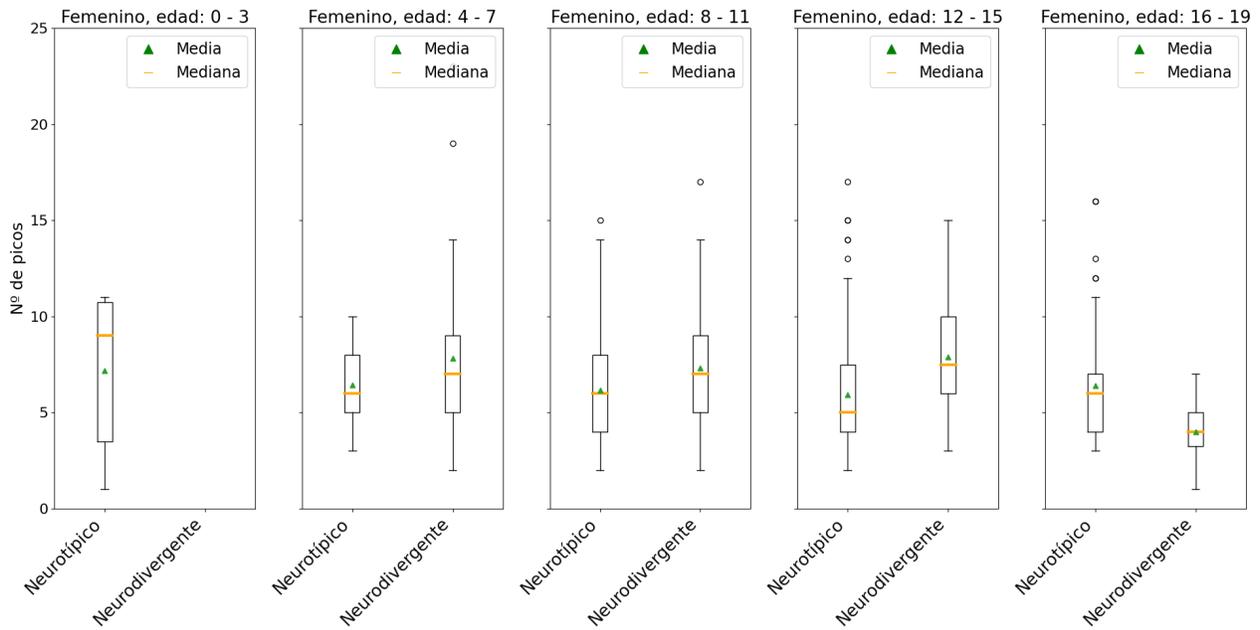


Ilustración 5.4: N^o de picos en niñas en la aceleración.

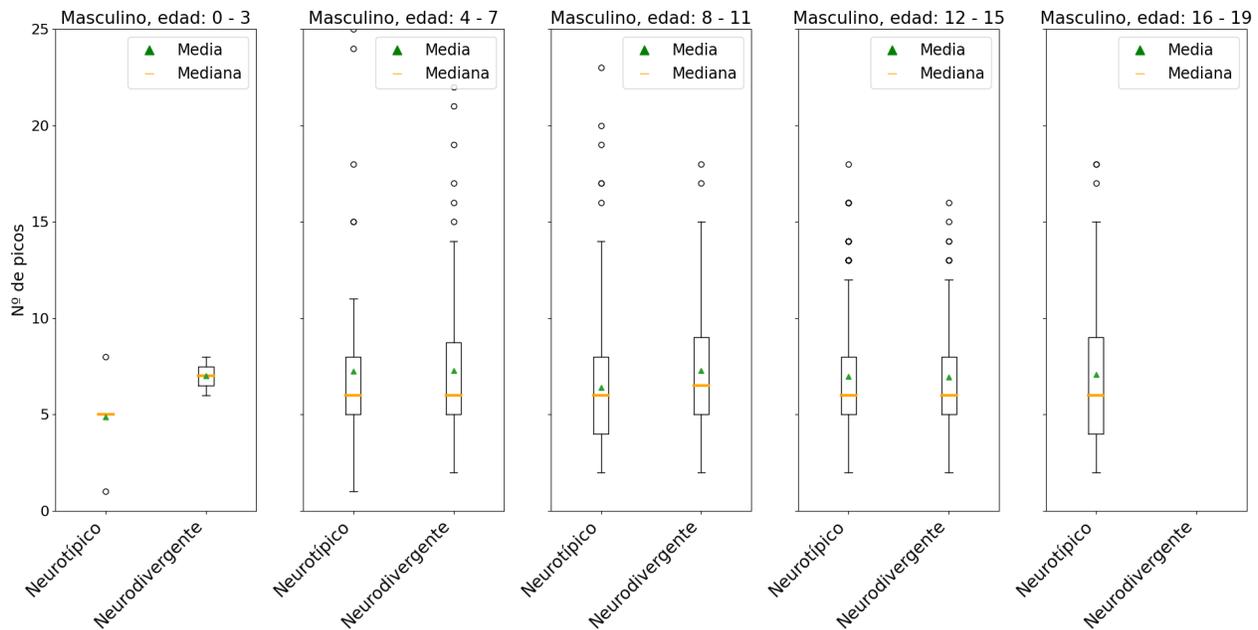


Ilustración 5.5: N^o de picos en niños en la aceleración.

- **Número de picos por tiempo de ejecución:**

Al analizar los resultados del algoritmo 5.6 y 5.7, se observó que no hay una tendencia clara en el número de picos por unidad de tiempo de ejecución, ya que los valores son muy parecidos y, en algunos casos, los valores son mayores para los niños neurotípicos y en otros para los niños neurodivergentes.

Teniendo en cuenta que en la prueba anterior el número de picos era mayor para los niños neurodivergentes, se planteó la hipótesis de que el aumento observado en los niños neurotípicos al considerar el tiempo de ejecución podría deberse a que estos completan los movimientos en menos tiempo, lo que incrementa la tasa de picos por unidad de tiempo.

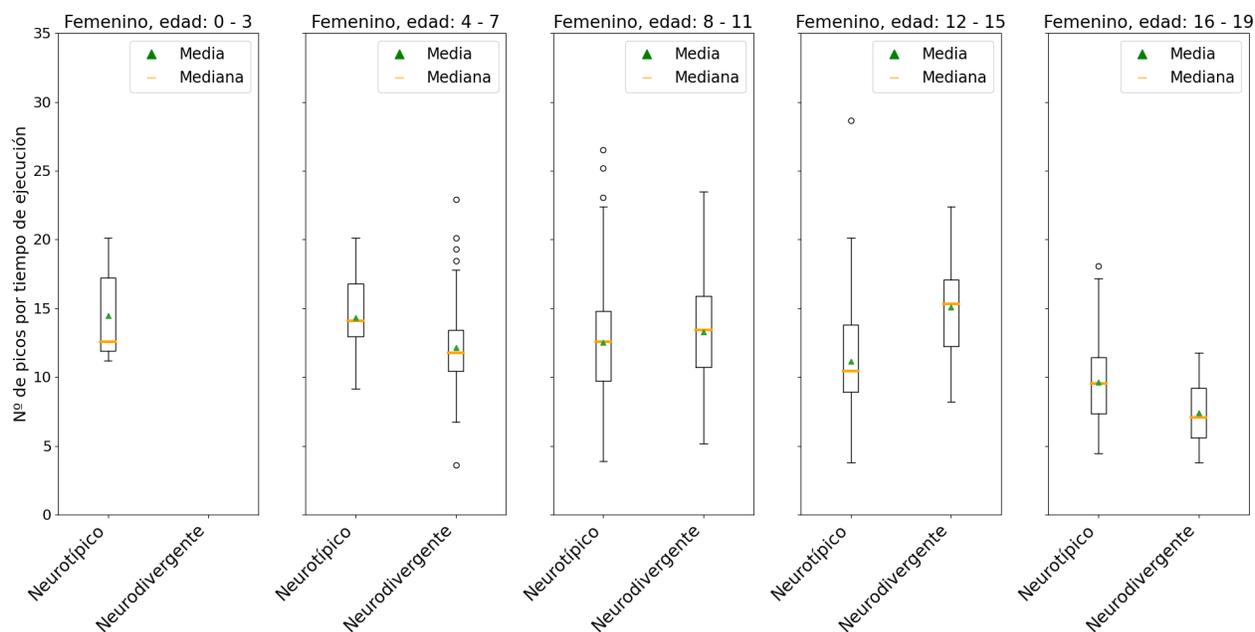


Ilustración 5.6: N° de picos por tiempo de ejecución en niñas en la aceleración.

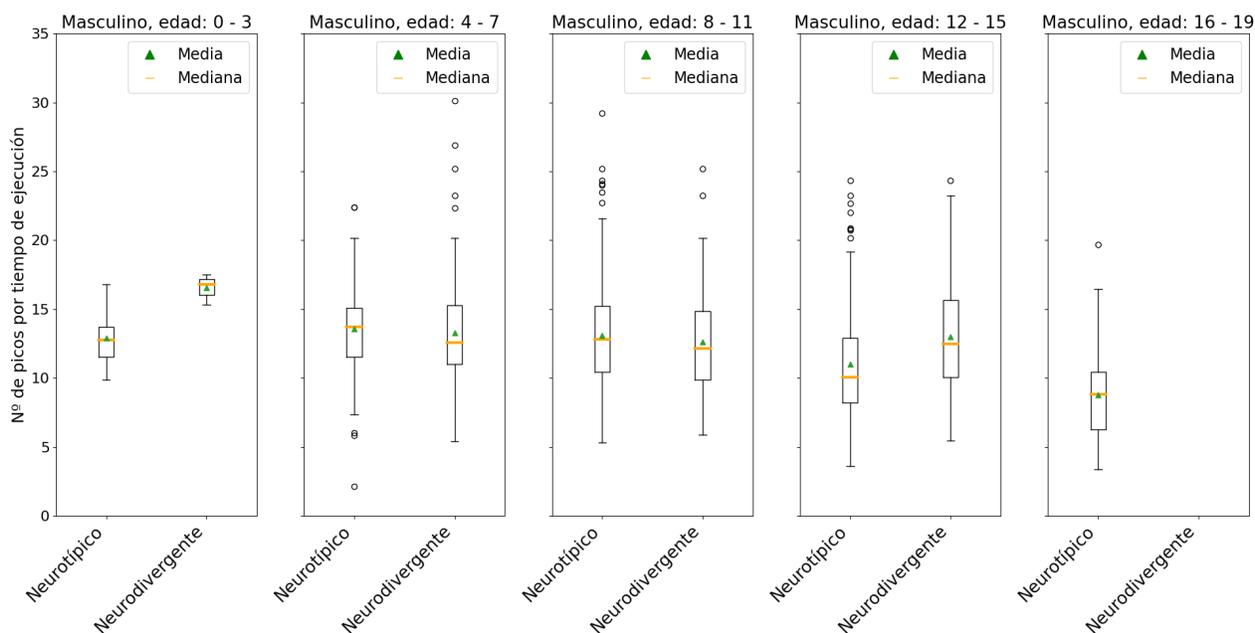


Ilustración 5.7: N° de picos por tiempo de ejecución en niños en la aceleración.

- **Tiempo total en realizar el ejercicio:**

Con este nuevo algoritmo, se buscó confirmar la hipótesis surgida del análisis del resultado anterior. Los resultados de los boxplot 5.8 5.9 indican que, en general, el tiempo de ejecución es menor para los niños neurotípicos, lo que sugiere que completan los movimientos más rápidamente en comparación con los niños neurodivergentes.

Esto podría explicar por qué los niños neurotípicos tienen una tasa de picos por unidad de tiempo más alta en algunos casos, ya que el menor tiempo de ejecución aumenta la frecuencia de picos detectados.

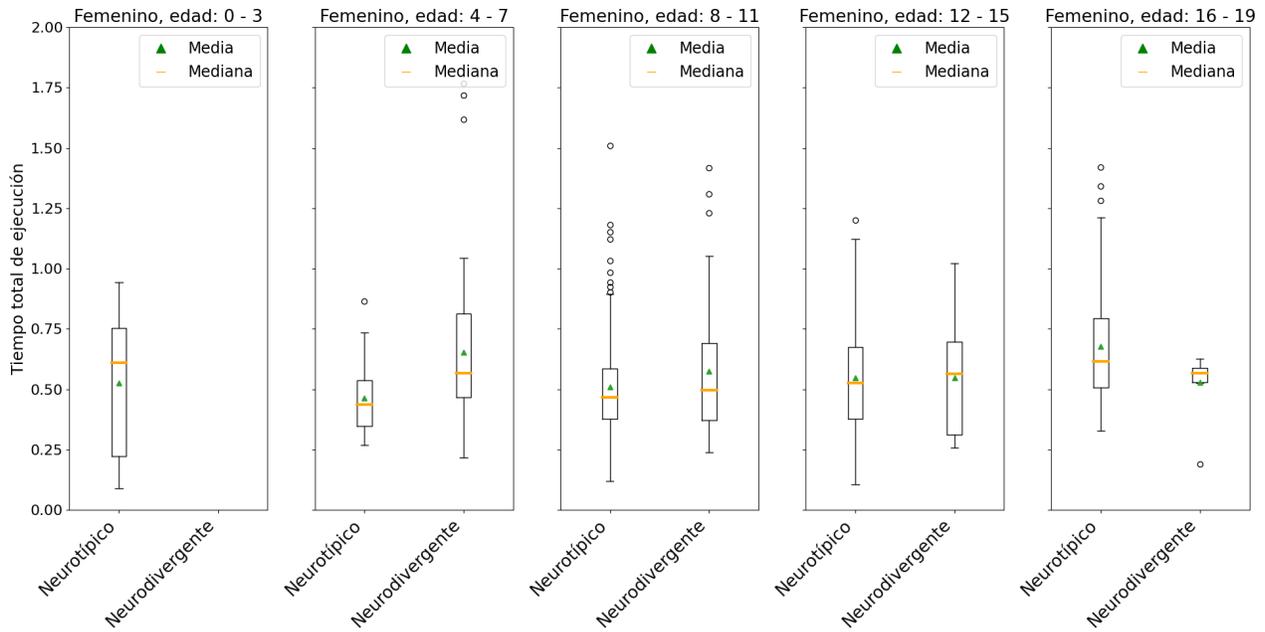


Ilustración 5.8: Tiempo total de ejecución en niñas.

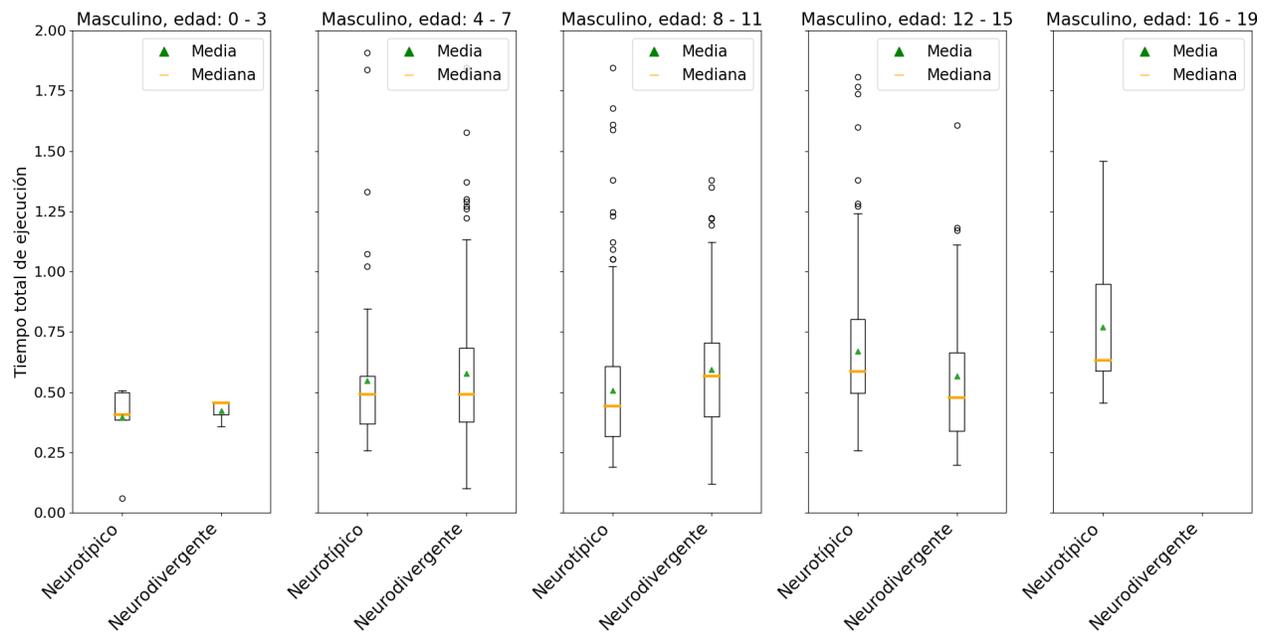


Ilustración 5.9: Tiempo total de ejecución en niños.

- **Tiempo desde que se empieza hasta que se alcanza la máxima velocidad:**

Se observa que las niñas y niños neurodivergentes alcanzan su máxima velocidad más rápidamente, como se muestra en las figuras 5.10 5.11. Además, en el caso de los niños neurodivergentes, se observa una alta dispersión de los resultados, lo que sugiere una mayor variabilidad en sus habilidades motoras. Esta mayor variabilidad puede indicar que, aunque algunos niños neurodivergentes pueden alcanzar rápidamente su máxima velocidad, otros pueden tardar más, reflejando diferencias individuales significativas en sus capacidades de coordinación y control motor.

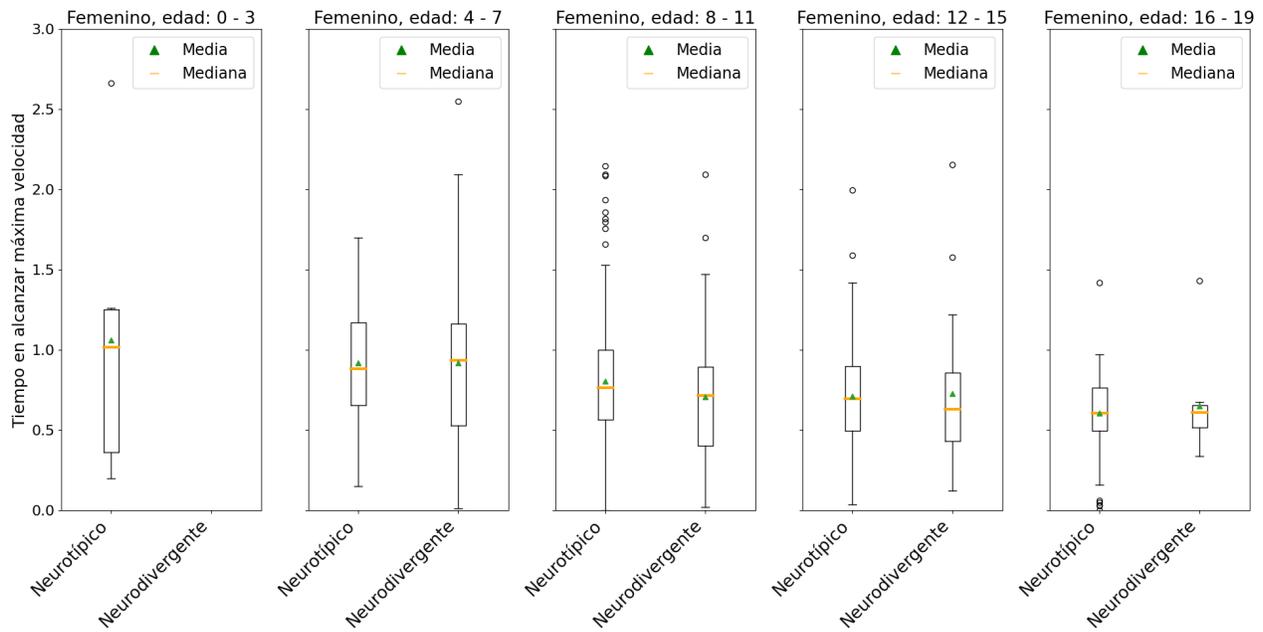


Ilustración 5.10: Tiempo que se tarda en alcanzar la máxima velocidad en niñas.

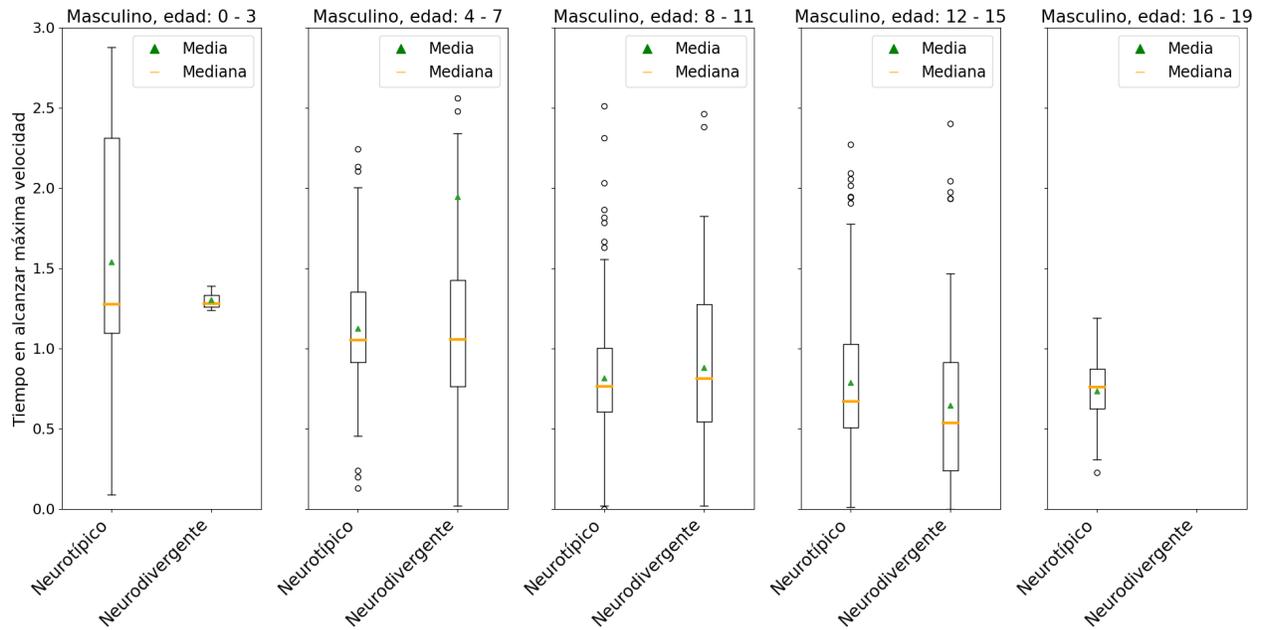


Ilustración 5.11: Tiempo que se tarda en alcanzar la máxima velocidad en niños.

- **Valor de la máxima velocidad alcanzada:** Se observa que los niños y niñas neurotípicos alcanzan una mayor velocidad máxima en comparación con los niños neurodivergentes 5.12 5.13.

Esta diferencia sugiere que los niños neurotípicos tienen la capacidad de generar movimientos más rápidos y posiblemente más controlados. En contraste, los niños neurodivergentes tienden a tener una menor velocidad máxima, lo que puede reflejar las dificultades en su capacidad para coordinar y ejecutar movimientos rápidos de manera eficiente.

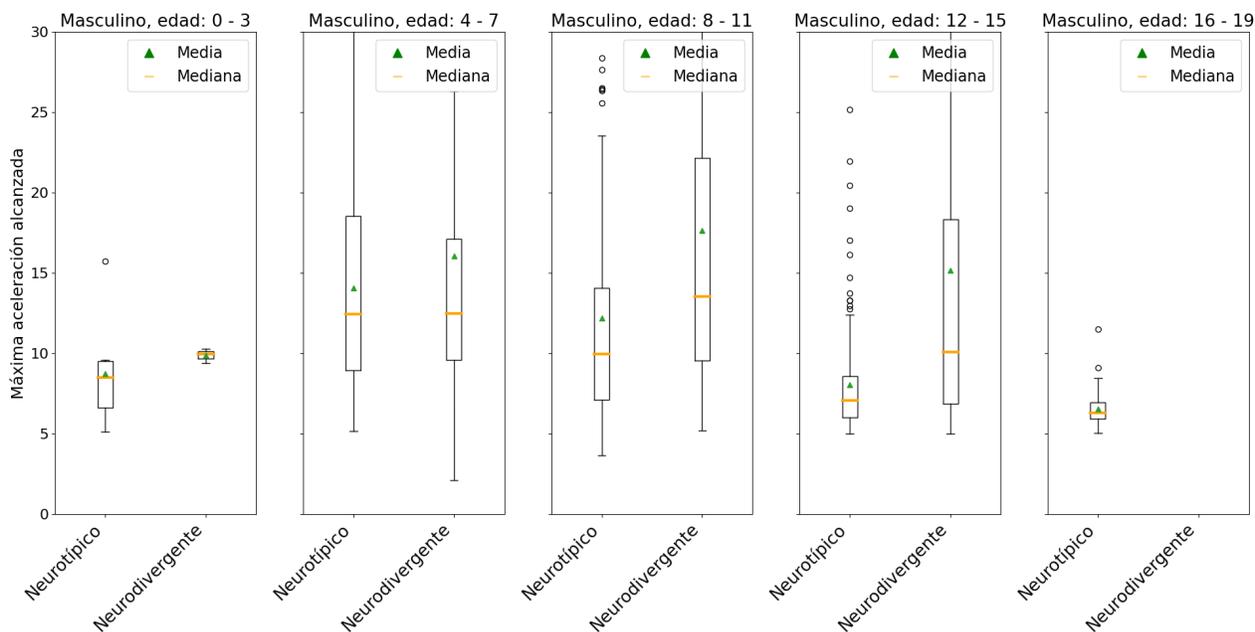


Ilustración 5.13: Máxima velocidad alcanzada en niños.

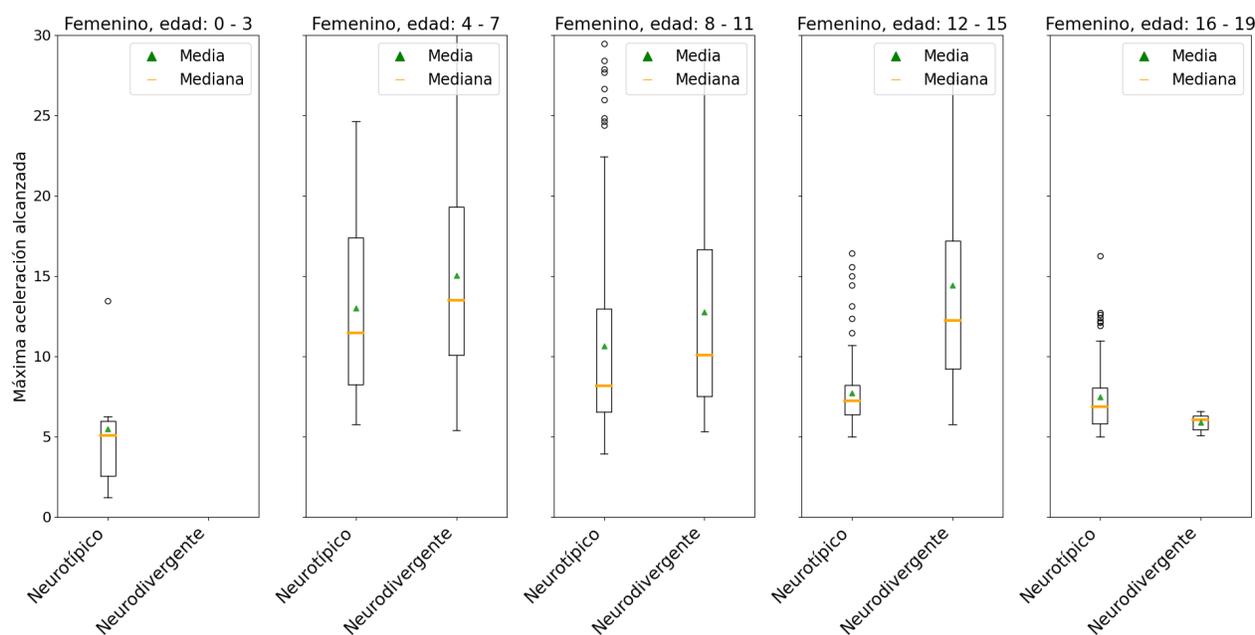


Ilustración 5.12: Máxima velocidad alcanzada en niñas.

Existe una tendencia clara en el aumento de esta velocidad en los niños neurodivergentes a medida que crecen. Además, hay una mayor dispersión de valores de velocidad máxima en los niños neurodivergentes, lo que sugiere una variabilidad significativa en sus habilidades motoras. En contraste, los niños neurotípicos muestran una menor dispersión, indicando una mayor consistencia en la velocidad máxima alcanzada.

5.3.4. Comparación de resultados algoritmos:

Después de desarrollar los algoritmos, se procedió a realizar un análisis para decidir qué método estadístico es más factible de aplicar, tales como ANOVA o Mann-Whitney. Se llevaron a cabo los siguientes análisis:

1. **Independencia:** La suposición de independencia se cumple para los datos evaluados. Esto significa que la observación de un individuo no influye en la observación de otro. Cada archivo se procesa de manera individual y los datos se han recolectado sin influencia mutua entre las observaciones. Esto asegura que las observaciones son independientes entre sí.
2. **Normalidad:** No se cumple para los datos evaluados. Se considera si los recuentos pueden no seguir una distribución normal. Para ello se usó la prueba Shapiro-Wilk que evalúa la hipótesis nula de que los datos provienen de una distribución normal.
3. **Homogeneidad de varianzas:** La mayoría de los grupos tienen varianzas homogéneas. Se utilizó la prueba de Levene para verificar la igualdad de las varianzas entre los grupos.

Debido a los resultados de estos análisis, se decidió emplear la prueba de Mann-Whitney U [20]. Esta es una prueba no paramétrica adecuada para comparar dos grupos independientes, ya que no asume normalidad, lo que la hace apropiada para analizar los datos.

Una vez se definió la prueba a utilizar, se realizó el cálculo del p-valor 5.3. El p-valor es una medida estadística que nos permite evaluar la evidencia en contra de la hipótesis nula. Un p-valor bajo (menor a 0.05) sugiere que la hipótesis nula puede ser rechazada, indicando que el parámetro estudiado por el algoritmo muestra diferencias significativas entre los grupos comparados [22]. Por lo tanto, un p-valor bajo apoyaría la hipótesis de que existen diferencias significativas entre los niños neurotípicos y neurodivergentes.

Parámetros	Femenino				Masculino		
	4-7	8-11	12-15	16-19	4-7	8-11	12-15
Nº de picos en la velocidad	0.6373	0.0298	0.0144	0.0015	0.5330	0.0686	0.6302
Nº de picos en la aceleración	0.1846	0.0100	0.0015	0.0070	0.5267	0.0081	0.9845
Nº de picos por tiempo en la velocidad	$2.42 \cdot 10^{-5}$	0.3221	0.0026	0.0510	0.2866	0.0124	$6.69 \cdot 10^{-5}$
Nº de picos por tiempo en la aceleración	0.0010	0.0790	$1.83 \cdot 10^{-5}$	0.0247	0.2046	0.2737	$2.35 \cdot 10^{-5}$
Duración total	0.0011	0.2237	0.9059	0.1815	0.4164	0.0002	0.0002
Tiempo hasta alcanzar la máxima velocidad	0.9311	0.0501	0.6345	0.8502	0.6251	0.3313	0.0023
Valor de la máxima velocidad alcanzada	0.0711	0.0055	$3.94 \cdot 10^{-7}$	0.0791	0.2142	$8.88 \cdot 10^{-7}$	$1.07 \cdot 10^{-9}$
Valor de la máxima aceleración alcanzada	0.1619	0.0067	$3.66 \cdot 10^{-6}$	0.0120	0.6213	$3.84 \cdot 10^{-7}$	$5.33 \cdot 10^{-8}$

Cuadro 5.3: Tabla con los algoritmos empleados y los resultados del P-valor por género y rango de edad

Se puede observar que los parámetros más significativos han sido el número de picos por tiempo en la aceleración y el valor de la máxima aceleración alcanzada, con 4 p-valores significativos en el primero y 5 en el segundo.

Número de picos por tiempo en la aceleración: Este parámetro muestra diferencias significativas en varias categorías de edad tanto en niñas como en niños. Estas diferencias sugieren que los niños neurodivergentes tienen movimientos menos precisos y controlados, lo cual se refleja en una mayor frecuencia de cambios en la aceleración durante sus movimientos.

Valor de la máxima aceleración alcanzada: Este parámetro también muestra diferencias significativas en varios grupos de edad tanto en niñas como en niños. En este caso, los niños neurodivergentes tienden a tener una mayor variabilidad en la máxima aceleración, lo que puede reflejar dificultades en generar movimientos rápidos y controlados, lo cual puede deberse a problemas de coordinación y fuerza.

5.3.5. Aplicación de redes neuronales:

Además de estudiar los parámetros que diferencian los dos casos de estudio, se ha entrenado una red neuronal utilizando las gráficas de aceleración. Aprovechando el trabajo anterior, se ha seleccionado solo la parte acotada donde se realiza el ejercicio, eliminando el ruido.

Lo primero fue generar las imágenes de la forma más homogénea posible estableciendo un límite en los ejes, igual para todos los casos. Para establecer el límite en el eje Y, se obtuvo la mayor aceleración alcanzada (valor del pico más alto) para cada género. En el caso de los niños, el valor más alto fue 88.06 m/s^2 , mientras que para las niñas, el valor más alto fue 37.17 m/s^2 , por lo tanto, se fijó el límite en 100. Una vez determinado el valor del eje Y, se estableció que el punto de inicio sea en el instante 0 para todos los casos, independientemente de en qué instante empezara el niño, y se acotó el tiempo total a 2 segundos. Una vez se tuvo todo correctamente acotado, se procedió a eliminar los ejes de la imagen para mejorar el entrenamiento de la red neuronal 5.15. Esta creación homogénea de las imágenes se automatizó creando un programa en *Python*.

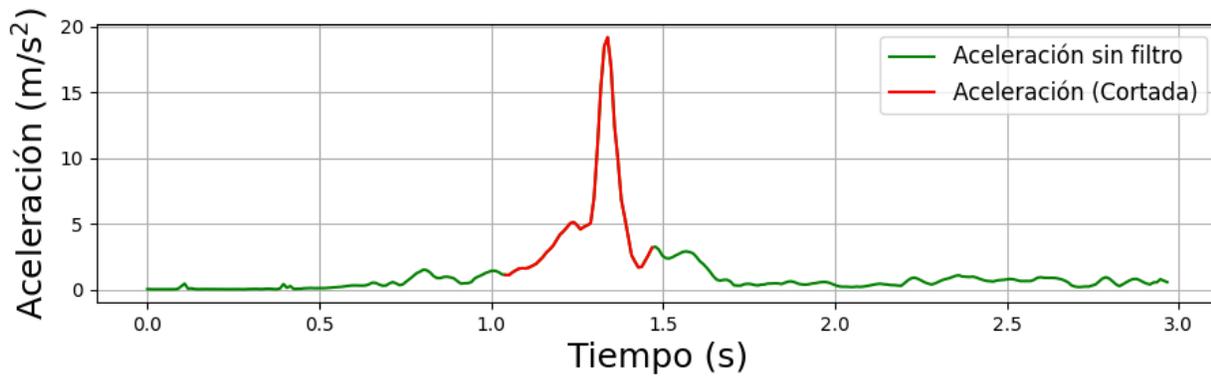


Ilustración 5.14: Antes de normalizar para su uso en el entrenamiento.

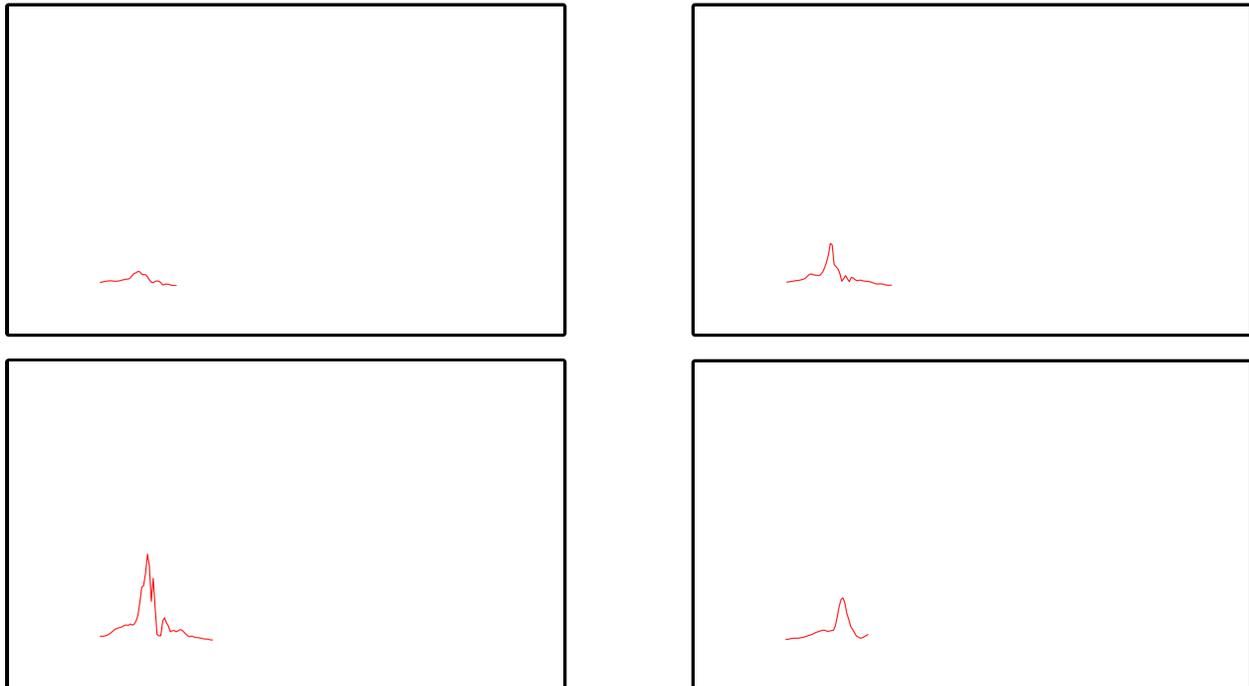


Ilustración 5.15: Ejemplos de gráficas normalizadas para usarlas en el entrenamiento.

Para entrenar correctamente la red neuronal, se dejaron solo tres gráficas para cada niño ya que varios niños solo habían hecho el ejercicio un máximo de 3 veces y para entrenar la red neuronal es necesario tener el mismo número de muestras para cada niño. Además, como hubo algunos casos con menos de 3 repeticiones, se dejaron fuera de la parte de entrenamiento y validación para usarlo para comprobar si clasifica bien.

Una vez el conjunto de imágenes estaba correcto, se entrenó la red neuronal tomando como base un código de *GitHub* [4]. Para entrenar la red neuronal, se modificaron varios hiperparámetros, los cuales son responsables de controlar el comportamiento del algoritmo y su configuración influye significativamente en el rendimiento del modelo. Además, estos se establecen antes del proceso de aprendizaje y no son ajustables durante el entrenamiento. A continuación, se describen los hiperparámetros empleados y el valor dado en el entrenamiento de la red neuronal:

1. **Tamaño del lote (*batch size*):** El *batch size* es un parámetro que permite establecer la cantidad de datos que se van a procesar antes de actualizar los parámetros del modelo. Para este proyecto se usó un *batch size* de 16, debido a las limitaciones de memoria. Sin embargo, este valor pequeño también ofrece la ventaja de que hace más actualizaciones de parámetros por época, aunque hace que los gradientes sean menos estables, lo que implica que los parámetros durante el entrenamiento puedan ser más erráticos o fluctuantes.
2. **Tasa de aprendizaje (*learning rate*):** Se ha utilizado el optimizador *Adam*, que emplea una tasa de aprendizaje adaptativa. Este optimizador ajusta dinámicamente la tasa de aprendizaje durante el entrenamiento basándose en las estimaciones de primer y segundo orden de los momentos (media y varianza) de los gradientes. Esto puede mejorar la estabilidad y eficiencia del proceso de optimización, adaptándose mejor a las características del problema en cuestión.
3. **Número de épocas (*epochs*):** Se han establecido 200 épocas. Sin embargo, esto no implica necesariamente que todas las épocas se completen. Se ha implementado una estrategia de parada temprana (*early stopping*) para monitorear la precisión en los datos de validación (*val_accuracy*). La parada temprana se activará si no se observa una mejora en la precisión de validación después de 15 épocas consecutivas (*patience=15*). Esta técnica ayuda a prevenir el sobreajuste y reduce el tiempo de entrenamiento innecesario.
4. **Función de activación:** Se ha usado *ReLU* para las capas convolucionales y densas intermedias. *ReLU* introduce no linealidad en el modelo y ayuda a evitar el problema del Gradiente desaparecido, permitiendo que la red aprenda de manera más eficiente. Para la capa de salida se ha utilizado *sigmoid*, que es apropiada para la clasificación binaria, ya que transforma las salidas en probabilidades dentro del rango $[0, 1]$, lo que facilita la interpretación de las clases previstas como probabilidades.
5. **Arquitectura de la red:** La estructura del modelo incluye varias capas:
 - **Capas Convolucionales (*Conv2D*):** Detectan características locales en las imágenes, como bordes y texturas.
 - **Capas de *Pooling* (*MaxPooling2D*):** Reducen la dimensionalidad y extraen

las características más importantes de los mapas de características.

- **Capas de *Dropout* (*Dropout*):** Ayudan a prevenir el sobreajuste desactivando aleatoriamente una fracción de neuronas durante el entrenamiento.
 - **Capa de Aplanamiento (*Flatten*):** Convierte el mapa de características multidimensional en un vector unidimensional para su procesamiento en capas densas.
 - **Capas Densas (*Dense*):** Integran y clasifican las características aprendidas, con la última capa usando *sigmoid*.
6. **Función de pérdida (*loss function*):** La función de pérdida utilizada es *categorical_crossentropy*, que es adecuada para problemas de clasificación multiclase. Mide la discrepancia entre las etiquetas verdaderas y las predicciones del modelo, penalizando más severamente las predicciones incorrectas y ayudando a mejorar la precisión del modelo.

El modelo de red neuronal convolucional (CNN) descrito está diseñado para clasificar imágenes de tamaño 600x1000 píxeles con 3 canales de color (RGB). La red comienza con una capa de rescalado (*Rescaling*) que ajusta los valores de los píxeles al rango $[-1, 1]$. A continuación, se aplican tres capas convolucionales (*Conv2D*) con activación *ReLU*, seguidas de capas de *max-pooling* (*MaxPooling2D*) para la reducción de la dimensionalidad. El tamaño de los filtros aplicados va variando según la red que se entrene.

Debido a los resultados significativos obtenidos en el grupo de edad de 12 a 15 años y de 8 a 11 años para ambos géneros en la obtención de parámetros distintivos entre los casos de estudio, se procedió a entrenar cuatro redes neuronales específicas para cada sexo dentro de este rango de edades.

1. Red neuronal 1: edad 8-11 años y género masculino.
2. Red neuronal 2: edad 8-11 años y género femenino.
3. Red neuronal 3: edad 12-15 años y género masculino.
4. Red neuronal 4: edad 12-15 años y género femenino.

Para cada red neuronal se va a usar un 70% de los datos para el entrenamiento y un 30% para la validación. Como se mencionó anteriormente para comprobar que funciona correctamente se van a usar las muestras de los niños que realizaron menos de 3 repeticiones. En la tabla 5.4 se indica a cuántas muestras corresponden estos porcentajes para cada caso.

Para cada red que se entrene, se ajustarán los parámetros clave para optimizar su rendimiento. Esto es, el número de filtros y el tamaño del *kernel* en las capas convolucionales, así como el número de neuronas en las capas densas y la tasa de *dropout*.

	Entrenamiento		Validación		Testeo	
	C	N	C	N	C	N
Red 1	66	27	30	15	10	8
Red 2	42	18	21	9	12	7
Red 3	33	30	18	15	9	6
Red 4	21	9	9	6	9	6

Cuadro 5.4: Resultados de las redes neuronales. (C: neurotípico; N: Neurodivergente)

5.3.6. Resultados red neuronal:

Al finalizar cada entrenamiento de cada red neuronal, se obtenía el informe de clasificación, una tabla que presenta los siguientes datos:

- ✓ **Precisión:** Se refiere a la proporción de instancias correctas sobre el total de instancias clasificadas como positivas por la red.
- ✓ **Sensibilidad:** Es la proporción de verdaderos positivos sobre el total de instancias que realmente son positivas.
- ✓ **Puntuación F1:** Es la media armónica de la precisión y la sensibilidad. Es una métrica más equilibrada que considera ambas.
- ✓ **Apoyo:** Indica el número de instancias verdaderas en cada clase.
- ✓ **Exactitud global:** La exactitud es la proporción de todas las instancias correctamente clasificadas sobre el total de instancias.

Una vez comprendidos los valores representados por cada parámetro del informe, se procede a evaluar y comparar los resultados obtenidos en el entrenamiento de cada red neuronal.

Red neuronal 1:

Cuadro 5.5: Resultados (%) de clasificación para la red neuronal 1

Clase	Precisión	Recall	F1-Score	Datos de validación
Neurodivergente	33.33	20.00 25.00	15	
Neurotípico	66.67	80.00	72.73	30
Exactitud	60.00			

El modelo 5.5 tiene un rendimiento muy poco preciso para los niños neurodivergentes. Esto se debe a que se dispone de muy pocos datos para este grupo en comparación con los que hay para los niños neurotípicos.

Para la clase neurodivergente, el modelo presenta una precisión de 33.33 en contraste la clase neurotípicos tiene una precisión de 80.00. Esto sugiere que la red puede llegar a obtener buenos resultados con los parámetros configurados si se entrena con mas datos.

En la matriz de confusión 5.16 se observa que el modelo ha clasificado casi todos los datos como neurotípicos confundiendo los datos.

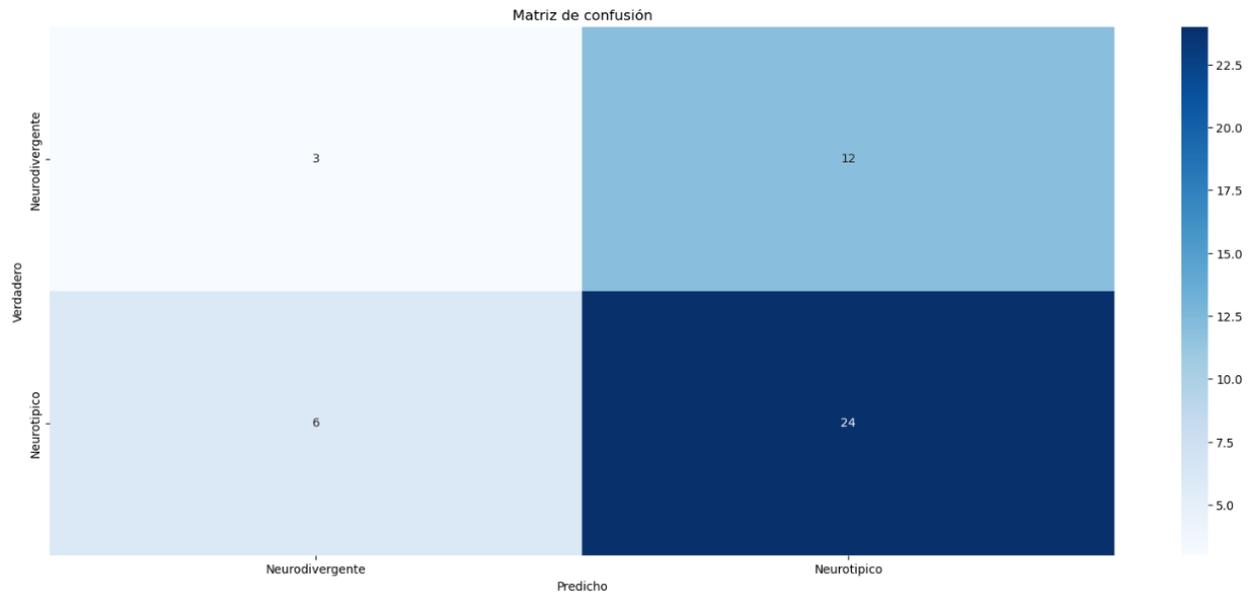


Ilustración 5.16: Matriz de confusión para la red 1.

Red neuronal 2:

Cuadro 5.6: Resultados (%) de clasificación para la red neuronal 2

Clase	Precisión	Recall	F1-Score	Datos de validación
Neurodivergente	62.50	55.56	58.82	9
Neurotípico	81.82	85.71	83.72	21
Exactitud	76.67			

El modelo 5.6 tiene un rendimiento moderado, con una exactitud general del 76.67%. Tiende a clasificar mejor las instancias neurotípicas en comparación con las neurodivergentes pero esto se debe en gran parte a la diferencia de datos.

Para la clase neurodivergente, el modelo presenta una precisión de 62.50 en contraste la clase neurotípicas tiene una precisión de 81.82. Esto sugiere que la red puede llegar a obtener buenos resultados con los parámetros configurados si se entrena con más datos.

En la matriz de confusión 5.17 se observa que el modelo solo ha clasificado correctamente 5 muestras de niños neurodivergentes y frente a 18 de niños neurotípicos. Asimismo, también hay 4 falsos positivos en el caso de neurodivergentes frente a 3 para los niños neurodivergentes.

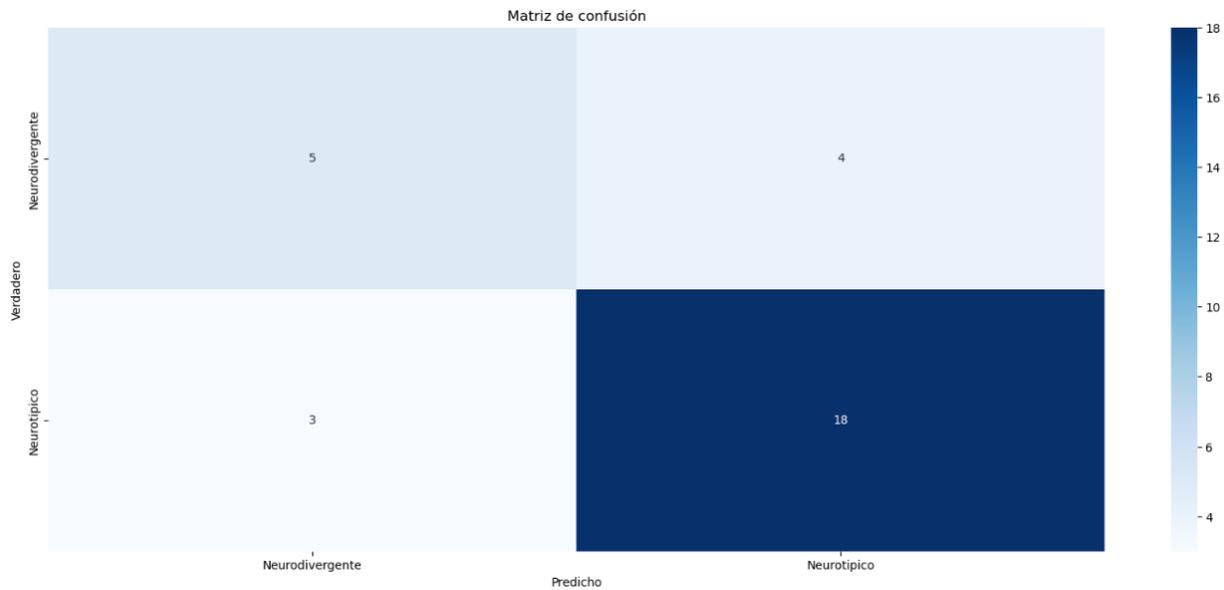


Ilustración 5.17: Matriz de confusión para la red 2.

Red neuronal 3:

Cuadro 5.7: Resultados (%) de clasificación para la red neuronal 3

Clase	Precisión	Recall	F1-Score	Datos de validación
Neurodivergente	75.00	80.00	77.42	15
Neurotípico	82.35	77.78	80.00	18
Exactitud	72.73			

El modelo 5.7 tiene un rendimiento moderado, con una exactitud general del 78.79%. Tiende a clasificar mejor las instancias neurotípicas en comparación con las neurodivergentes, como se evidencia por las métricas de precisión, *recall* y *F1-score* que son ligeramente más altas para la clase neurotípica.

Los errores de clasificación indican que el modelo puede necesitar ajustes o más datos para mejorar su capacidad de distinguir entre clases, especialmente para reducir los falsos negativos y falsos positivos en la clase neurodivergente.

En la matriz de confusión 5.18 se observa que el modelo ha clasificado correctamente 12 muestras de niños neurodivergentes y 14 de niños neurotípicos. Asimismo, también hay 3 falsos positivos en el caso de neurodivergentes frente a 4 para los niños neurodivergentes.

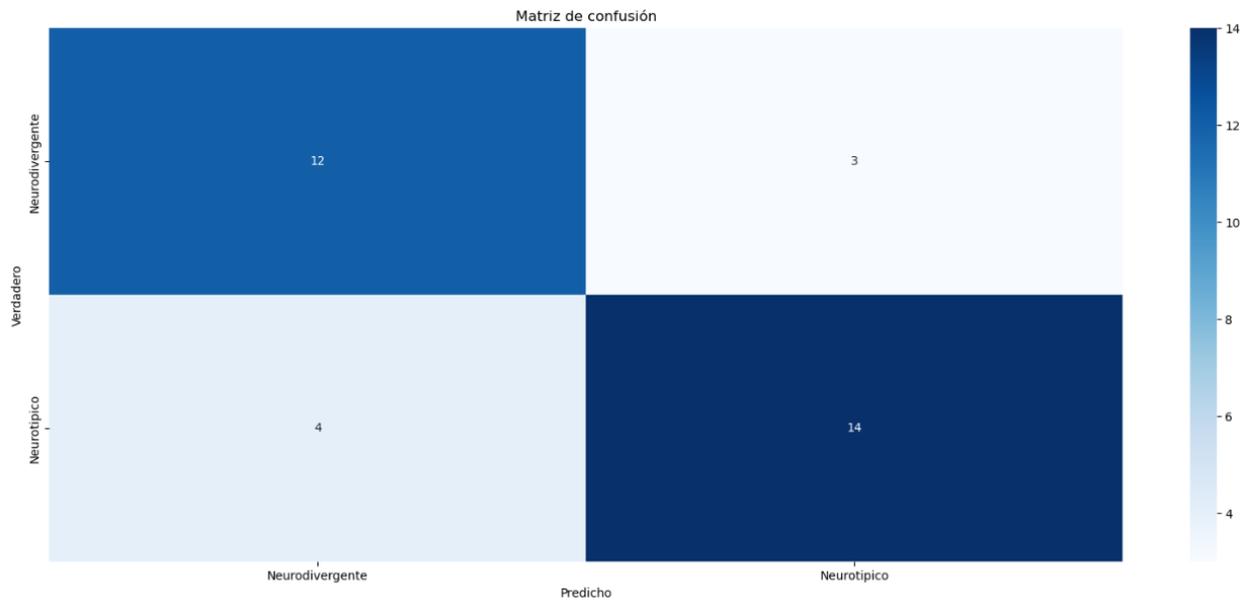


Ilustración 5.18: Matriz de confusión para la red 3.

Red neuronal 4:

Cuadro 5.8: Resultados (%) de clasificación para la red neuronal 4

Clase	Precisión	Recall	F1-Score	Datos de validación
Neurodivergente	100.00	16.67	28.57	6
Neurotípico	64.29	100.00	78.26	9
Exactitud	66.67			

El modelo 5.8 tiene una alta precisión pero un bajo *recall* para la clase Neurodivergente, lo que indica que solo predice Neurodivergente cuando está muy seguro, perdiendo muchas instancias verdaderas de Neurodivergente. Para la clase Neurotípico, el modelo funciona bien con un *recall* perfecto pero una precisión moderada, lo que significa que identifica correctamente todas las instancias de Neurotípico, pero a veces clasifica erróneamente a los Neurodivergentes como Neurotípicos.

En la matriz de confusión 5.19 se observa que el modelo ha clasificado correctamente 12 muestras de niños neurodivergentes y 14 de niños neurotípicos. Asimismo, también hay 3 falsos positivos en el caso de neurodivergentes frente a 4 para los niños neurodivergentes.

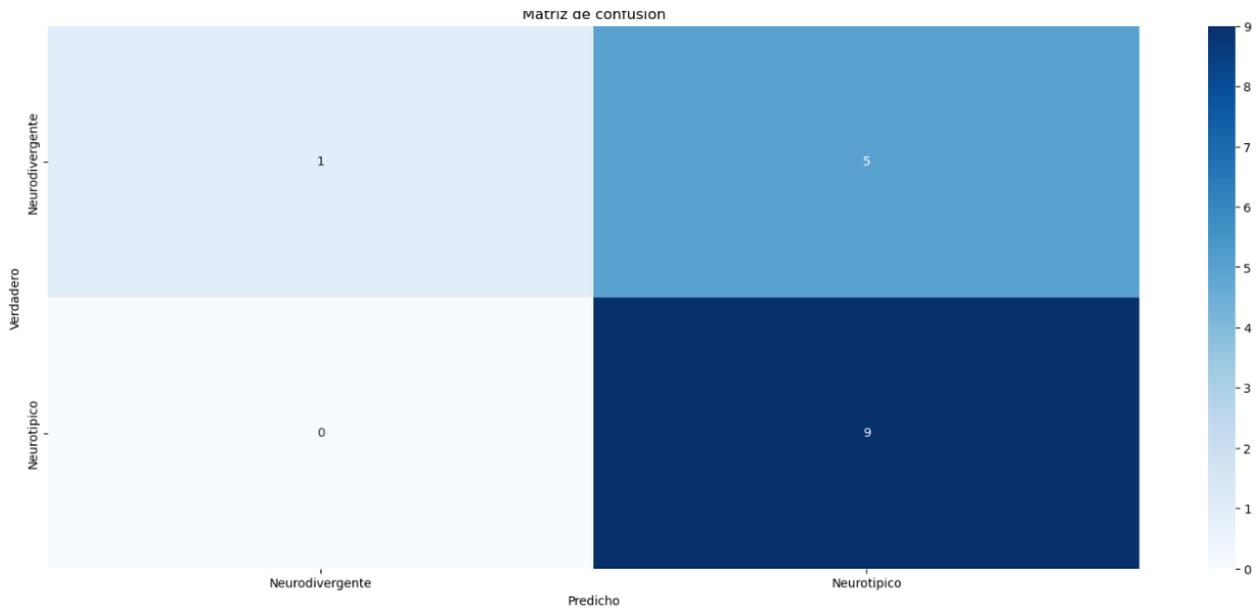


Ilustración 5.19: Matriz de confusión para la red 4.

5.3.7. Discusión:

Los p-valores significativos en el valor del pico más alto, la duración total y el número de picos por tiempo reflejan las diferencias en el control motor y la coordinación entre los niños neurotípicos y los neurodivergentes.

En el caso del pico más alto, los niños con problemas en el neurodesarrollo tienen dificultades para realizar movimientos de manera óptima. Esto resulta en mayores velocidades y aceleraciones durante el movimiento, reflejándose en picos más altos en las mediciones de velocidad y aceleración.

Con respecto a la duración total del movimiento, los problemas de sincronización comunes en los niños neurodivergentes provocan que tarden más tiempo en completar un movimiento en comparación con los niños neurotípicos.

Finalmente, en cuanto al número de picos por unidad de tiempo, los niños neurodivergentes no son capaces de realizar movimientos tan precisos y controlados como los niños neurotípicos. Esta imprecisión en el movimiento se traduce en un mayor número de picos, ya que sus movimientos tienden a ser más erráticos y menos suaves.

Estos resultados apoyan la hipótesis de que existen diferencias significativas en los patrones de movimiento entre ambos grupos, lo cual es consistente con las dificultades motoras y de sincronización observadas en los niños con trastornos del neurodesarrollo.

Los resultados obtenidos para la red neuronal son prometedores cuando se dispone de suficientes datos y no hay una diferencia muy grande entre los datos de ambas clases. Para mejorar los resultados obtenidos con la red neuronal, es fundamental aumentar la cantidad de datos disponibles, especialmente los datos de niños neurodivergentes, ya que existe una

significativa disparidad en comparación con los datos de niños neurotípicos. Al lograr este objetivo, se espera que la red neuronal tenga un rendimiento más equilibrado y preciso. Además, los resultados mejorados de la red neuronal proporcionarían un valioso apoyo a los especialistas, facilitando un diagnóstico más preciso y temprano de los niños neurodivergentes.

5.4. Limitaciones de la investigación:

Durante la toma de datos con niños se encontraron algunas dificultades que afectaron la calidad de las grabaciones. En esta sección, se describen los problemas encontrados y se plantean nuevos métodos para mejorar la calidad de los datos y evitar la necesidad de eliminar grabaciones incorrectas obteniendo un proceso más eficiente.

El primer problema identificado fue que algunos niños no comprendían qué debían hacer. Para abordar este problema, se propone realizar las pruebas con dos niños de diferentes edades. La idea es que el niño más pequeño pueda imitar los movimientos del mayor, quien ya comprende bien la tarea. Esta técnica puede facilitar el entendimiento a través de la observación y la imitación.

El segundo problema surgió cuando algunos niños, a pesar de entender la tarea, realizaban el ejercicio con la mano que no llevaba el reloj, incluso después de ser instruidos a usar la mano con el reloj. Una posible solución es asignarles una tarea específica para la mano sin reloj, como dar la mano o sujetar un objeto. Esto ayuda a evitar que cambien de mano el objeto del ejercicio.

El tercer problema fue la ejecución del ejercicio fuera de tiempo, ya sea antes de empezar a grabar o después de haber terminado la grabación. Este problema fue particularmente notable en el movimiento de lanzar, que solo tiene 3 segundos de grabación. Una solución sencilla es aumentar el tiempo de grabación a 5 segundos. Actualmente se cuenta con un algoritmo que recorta los datos y obtiene únicamente el periodo en que se realizó el movimiento, por lo que añadir unos segundos extra no debería representar un inconveniente y puede mejorar la precisión del registro.

Capítulo 6

Conclusiones y trabajo futuro

Se ha cumplido con los objetivos principales del proyecto. A través del estudio realizado sobre los parámetros de movimiento, se han identificado diferencias significativas entre niños neurotípicos y niños neurodivergentes. A continuación, se comentan los logros de este proyecto.

- **Desarrollo y validación de algoritmos:** Se han desarrollado y validado algoritmos capaces de medir y analizar parámetros en la aceleración y en la velocidad. Esto ha permitido encontrar parámetros significativos que diferencien las habilidades motoras entre niños neurotípicos y niños neurodivergentes.
- **Identificación de parámetros significativos:** Los valores estadísticos obtenidos demuestran que el número de picos por tiempo en la aceleración y el valor de la máxima aceleración alcanzada son parámetros particularmente útiles para diferenciar entre niños neurotípicos y niños neurodivergentes, indicando que hay diferencias significativas en los patrones de movimiento.
- **Obtención de datos para niños entre 0 y 3 años:** Si bien no se ha obtenido una gran cantidad de datos para estas edades, si se han podido identificar las dificultades que conlleva realizar las pruebas con los niños en edades tan tempranas. Esto facilitará la obtención de datos no erráticos en un futuro.

En resumen, se ha conseguido obtener varios métodos para diferenciar entre niños neurotípicos y neurodivergentes. Sin embargo no es del todo eficaz debido a la falta de datos para tener resultados precisos.

6.1. Trabajo futuro

En este apartado se van a especificar los pasos a seguir en el futuro para conseguir mayores avances y mejores resultados en este trabajo de investigación.

- **Ampliación de la base de datos:** Es necesario recolectar más datos para confirmar y ampliar los hallazgos actuales. Un conjunto de datos más grande permitirá una validación más robusta y la posibilidad de encontrar nuevos patrones y diferencias.
- **Desarrollo de interfaz gráfica:** Para ayudar a los profesionales de la salud, sería muy útil disponer de una interfaz gráfica que permita visualizar los resultados de los movimientos de los niños y compararlos con los de niños neurotípicos y neurodivergentes. Esta herramienta facilitaría la evaluación de la condición de neurodesarrollo, proporcionando una referencia visual clara y comprensible.
- **Analizar el resto de ejercicios:** Usando como referencia este proyecto, y aprovechando los algoritmos creados, se podrían aplicar para el resto de ejercicios grabados. Esto ayudaría a tener más variabilidad en la realización de las pruebas por parte de los niños ya que pueden no querer realizar el ejercicio de lanzar.

Lo principal es obtener más datos para conseguir evaluar de forma más precisa el resultado obtenido con los datos que se tienen hasta el momento.

Cuando se disponga de estos datos, hay que aplicarlos para el estudio de los parámetros y para entrenar a la red neuronal, consiguiendo que dé resultados más precisos.

Además, sería de gran ayuda a los especialistas que se creara una interfaz gráfica con la que puedan ver el resultado obtenido del niño y compararla con la de un niño neurotípico además de ver dónde se encuentra el niño dentro del boxplot.

Capítulo 7

Bibliografía

Bibliografía

- [1] Anaconda, Inc. (2024). Anaconda Assistant.
- [2] Apple (2024). TestFlight.
- [3] Aprende Machine Learning (2018). ¿Cómo funcionan las Convolutional Neural Networks? Visión por Ordenador.
- [4] Artal, C. G. (2024). Reconocedor de lengua de signos.
- [5] Atlassian Corporation (2011). Trello.
- [6] Barrero García, Noelia; San Miguel Lomo, Marta; Díaz Díaz, Noelia; Díaz Cáceres, Elena. (2019). Trastorno por déficit de atención e hiperactividad: Habilidades motoras en actividades de la vida diaria. Un estudio piloto. Revista médica.
- [7] Conners, K. (2007). Conner’s comprehensive behavior rating scalesTM. *Multi-Health Systems Inc.*, 1:2.
- [8] Digital Science (2024). Overleaf.
- [9] Escuela de Ingeniería Informática Universidad de Las Palmas de Gran Canaria (2019). Grado en Ingeniería Informática.
- [10] Fitzgerald, M. (2018). The clinical utility of the ADI-R and ADOS in diagnosing autism. Cambridge University Press.
- [11] for Autism, L. C. (2007-2024a). What is the ADI-R (Autism Diagnostic Interview-Revised)? Massachusetts General Hospital, Founding Member, Mass General Brigham.
- [12] for Autism, L. C. (2007-2024b). What is the ADOS (Autism Diagnostic Observation Schedule)? Massachusetts General Hospital, Founding Member, Mass General Brigham.
- [13] Fortea-Sevilla, S., García-Sosa, A., Morales-Almeida, P., and Carmona-Duarte, C. (2023). Assessment of autism and adhd: A comparative analysis of drawing velocity profiles and the nepsy test. In *21th Conference of the International Graphonomics Society (IGS2023)*.
- [14] GitHub, Inc (2024). GitHub.

- [15] Gobierno de España. Ministerio de la presidencia justicia y relaciones con las cortes (2008). Ley orgánica 3/2018, de 5 de diciembre, de protección de datos personales y garantía de los derechos digitales.
- [16] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- [17] OpenAI (2024). Chatgpt. <https://www.openai.com/chatgpt>.
- [18] Itai Berger, Ortal Slobodin, H. C. (2017). Usefulness and Validity of Continuous Performance Tests in the Diagnosis of Attention-Deficit Hyperactivity Disorder Children. Oxford University Press.
- [19] Jet Brains (2000-2024). PyCharm.
- [20] Joaquín Amat Rodrigo (2017). Test de Wilcoxon Mann Whitney como alternativa al t-test.
- [21] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- [22] Molina Arias M. (2017). ¿Qué significa realmente el valor de p?
- [23] PcComponentes (2024a). Apple iPhone SE 2022 64GB (PRODUCT) RED Libre.
- [24] PcComponentes (2024b). Apple Watch Series 7 GPS + Cellular 41mm Aluminio Negro con Correa Deportiva Negra.
- [25] PcComponentes (2024c). HP OMEN 16-wd0011ns Intel Core i7-13620H/32GB/1TB SSD/RTX 4060/16.1”.
- [26] Python Software Foundation (2001-2024). Python.
- [27] Scott Chacon, Jason Long (2024). Git.
- [28] Sharma, A., Arora, S., Kumar, S., and Bansal, A. (2023). Ai ml enabled wearable smart sensors detecting psychological disorders. In *2023 13th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pages 651–656.
- [29] Shin, J., Maniruzzaman, M., Uchida, Y., Hasan, M. A. M., Megumi, A., and Yasumura, A. (2023). Handwriting-based adhd detection for children having asd using machine learning approaches. *IEEE Access*, 11:84974–84984.
- [30] Silvia Lombart Conesa, Patricia Fumanal Lacoma, Ester Ibort Torres, Jara Pisa Mañas, Isabel Fernández Sasal, Fernando Bandrés Latorre (2023). Alteraciones motoras del niño con trastorno del espectro autista: actualización del tratamiento en fisioterapia. *Revista Sanitaria de Investigación*.
- [31] The Jupyter Trademark (2024). Jupyter Notebook.

Glosario

gradiente desaparecido El problema del gradiente desaparecido ocurre cuando los gradientes utilizados para actualizar los pesos de una red neuronal se vuelven extremadamente pequeños a medida que se propagan hacia atrás desde las capas de salida hacia las capas de entrada durante el proceso de retropropagación. Esto puede hacer que las capas iniciales aprendan muy lentamente o dejen de aprender por completo. 36

Apéndice A

Normativa y legislación

Para el desarrollo de este proyecto, se han obtenido datos personales de niños, específicamente su edad, género y datos médicos relativos al neurodesarrollo. Dado que se trata de información personal, se ha cumplido con la “*Ley Orgánica 3/2018, de Protección de Datos Personales y garantía de los derechos digitales*” (LOPDyGDD)[15], asegurando el tratamiento adecuado de los datos personales y respetando los derechos de los interesados, en este caso, los padres, madres o tutores legales de los niños.

Para cumplir con la normativa vigente, se ha proporcionado información clara y detallada sobre la finalidad de la recogida de los datos. Asimismo, se entregó un documento de consentimiento informado a los interesados, que debieron completar y firmar, autorizando explícitamente el tratamiento de los datos de los menores.

Con estas acciones, se garantiza la transparencia y el cumplimiento de la legislación vigente en materia de protección de datos, asegurando que los datos personales de los niños sean tratados de manera segura y respetuosa con sus derechos y los de sus representantes legales.

Apéndice B

Presupuesto

En este apéndice se detalla el costo que supondría realizar este trabajo en caso de haber sido desarrollado para una empresa.

- **Costes de Software** El software empleado en este proyecto ha sido todo software libre, por lo que no supondría ningún coste adicional.
- **Costes de Hardware** Se han utilizado varios dispositivos de hardware. Dado que el hardware empleado ya no se encuentra comúnmente en el mercado, se ha considerado su equivalente más cercano disponible en el mercado sin ser de segunda mano.

Producto	Precio
HP OMEN 16-wd0011ns [25]	1.599,00€
iPhone SE [23]	469,00€
Apple Watch Series 7 [24]	399,00€
Total	2.167,00€

Cuadro B.1: Tabla de precios de productos hardware

- **Costes de personal** Teniendo en cuenta que el salario bruto anual de un programador junior oscila entre los 17.000 y 22.000 €, se ha tomado como referencia un valor promedio de 19.000 €. Considerando que se han dedicado aproximadamente 300 horas al TFG, lo que corresponde aproximadamente a dos meses de trabajo, el costo de personal sería en total 3166,66 €.
- **Resumen de Costes**

Concepto	Coste
Hardware	2.467,00 €
Personal	3.167,00 €
Total	5.634,00 €

Cuadro B.2: Resumen de costos del proyecto