



SISTEMA DE REHABILITACIÓN BASADO EN ANÁLISIS BIOMECÁNICO Y VISIÓN ESTÉREO

Integrantes

José de Jesús Fernández García

Jorge Enrique González Sevilla

Índice

Contenido

Introducción	4
Metodología	6
Regresión	6
Clasificación	11
Tipos de posturas a clasificar	11
Obtención de los ángulos de Euler para la clasificación	11
Limpieza de datos	14
Creación del conjunto de datos	17
Visualización de los datos	17
División del conjunto de datos (entrenamiento y prueba)	19
Selección de los mejores hiperparámetros para el modelo de clasificación binaria ...	20
Resultados	25
Clasificación	25
F1 Score de cada combinación de hiperparametros	25
Resultados con el conjunto de prueba	28
Regresión	31
Sistema informático	33
Discusión y conclusiones	36
Regresión	36
Clasificación	36
Conclusión general	37
Referencias	38

Introducción

La integración de tecnologías emergentes como la inteligencia artificial (IA) en la rehabilitación física está transformando significativamente el paradigma tradicional, ofreciendo soluciones innovadoras y efectivas en la evaluación y tratamiento de diversas patologías musculoesqueléticas. El presente proyecto tiene como objetivo aplicar técnicas de machine learning en el ámbito de la rehabilitación física para evaluar con precisión los movimientos corporales y proporcionar retroalimentación objetiva a los pacientes y especialistas en salud.

La rehabilitación física se enfrenta a diversos desafíos, especialmente relacionados con la evaluación precisa y objetiva del desempeño del paciente durante las sesiones terapéuticas. Los métodos convencionales suelen depender del juicio subjetivo del terapeuta, lo cual puede dificultar un seguimiento adecuado del progreso y limitar la efectividad de los tratamientos implementados. Frente a esta problemática, se propone desarrollar un sistema basado en visión estéreo utilizando dos cámaras USB económicas que permitirán una captura detallada y una evaluación precisa de los movimientos corporales en tres dimensiones, utilizando técnicas avanzadas de machine learning.

El fundamento del sistema radica en la técnica de visión estéreo, que imita la percepción humana mediante triangulación desde múltiples puntos de vista, permitiendo una percepción precisa de la profundidad espacial. Este sistema se complementa con modelos avanzados de deep learning que facilitan la identificación automática de puntos anatómicos clave para la evaluación biomecánica precisa.

Diversas investigaciones han validado el potencial de estas tecnologías para aplicaciones clínicas en rehabilitación física. Por ejemplo, Muñoz et al. (2013) demostraron cómo el sensor Kinect combinado con videojuegos interactivos puede mejorar la experiencia del paciente, ofreciendo simultáneamente una evaluación objetiva de los movimientos mediante análisis biomecánico automatizado. Por su parte, Uhlich et al. (2022) desarrollaron OpenCap, un sistema abierto que permite capturar movimientos corporales mediante smartphones, validando su precisión y accesibilidad en contextos clínicos y domiciliarios.

Adicionalmente, investigaciones recientes han aplicado técnicas de machine learning para mejorar aún más la precisión y personalización de los tratamientos. Keller et al. (2022) utilizaron aprendizaje no supervisado para identificar patrones de movimiento distintivos en pacientes con dolor lumbar crónico, logrando diferenciar estos patrones de individuos sanos. De manera similar, Phan et al. (2024) aplicaron modelos de regresión basados en aprendizaje automático para predecir cambios en patrones de movimiento durante la

rehabilitación, demostrando la utilidad clínica de estas técnicas para optimizar estrategias terapéuticas.

Liao et al. (2020) propusieron un marco basado en redes neuronales profundas que permite evaluar automáticamente la calidad de ejercicios de rehabilitación física, proporcionando una evaluación objetiva y consistente del rendimiento del paciente. Por su parte, Obukhov et al. (2023) examinaron diferentes sistemas de seguimiento del movimiento, incluyendo visión por computadora y sensores inerciales, para determinar su precisión en aplicaciones clínicas, concluyendo que estas tecnologías pueden integrarse eficazmente en la monitorización y evaluación de ejercicios terapéuticos.

Finalmente, el trabajo de Zago et al. (2020) mostró la viabilidad y precisión de sistemas basados en visión estéreo de bajo costo para el seguimiento del movimiento corporal tridimensional, lo que confirma la idoneidad de esta tecnología para aplicaciones clínicas accesibles y efectivas.

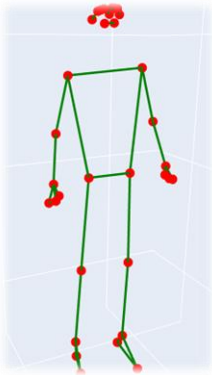
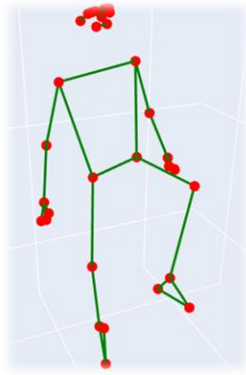
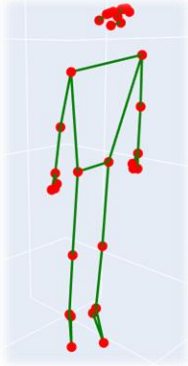
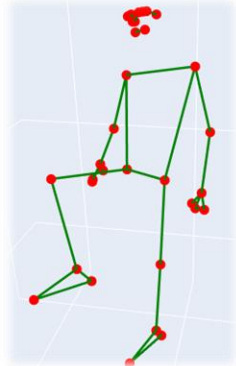
Metodología

Regresión

Debido a la naturaleza de los datos para el problema de regresión, tiene más sentido analizar visualmente la nube de puntos de cada punto 3D a lo largo del tiempo.

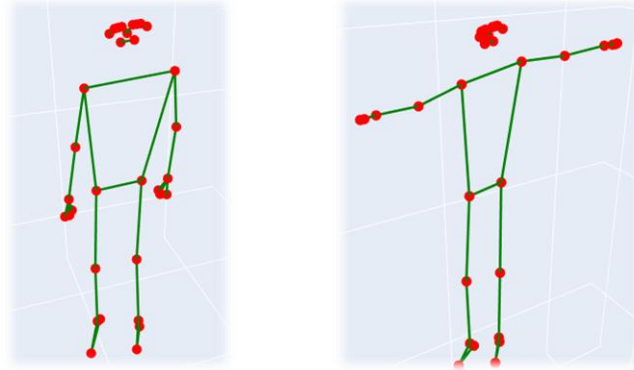
Las trayectorias para analizar son las siguientes:

- Levantamiento de rodilla izquierda
- Levantamiento de rodilla derecha
- Levantamiento de brazos

Nombre	Postura inicial de la trayectoria	Postura final de la trayectoria
<p><i>Levantamiento de rodilla izquierda</i></p> <p>Numero de datos antes de la limpieza: 41 Numero de datos después de la limpieza: 26</p>		
<p><i>Levantamiento de rodilla derecha</i></p> <p>Numero de datos antes de la limpieza: 47 Numero de datos después de la limpieza: 29</p>		

Levantamiento de brazos

Numero de datos antes de la
limpieza: 60
Numero de datos después de la
limpieza: 60



La limpieza de las posturas (para no agregar ruido) se realiza utilizando un método visual, la cual consiste en ir postura por postura analizando que no existan posturas fuera de lo normal, y si es que existen se eliminan.

El proceso de limpieza no es un proceso muy tardado, ya que una trayectoria a lo mucho puede tener 50 posturas diferentes.

Por lo pronto se utiliza un método visual y manual, pero podría ser de utilidad encontrar un método automático que ayude a eliminar posturas ruidosas a lo largo de la trayectoria.

Función objetivo a minimizar para el ajuste de trayectoria de cada punto 3D:

$$F = \frac{1}{2m} \sum_{i=1}^m \|u^{(i)} - p(t^{(i)})\|^2$$

Donde:

- m : Numero de puntos 3D en la nube de puntos que describe una trayectoria
- $p(t^{(i)})$: Un punto sobre la trayectoria ajustada
- $u^{(i)}$: Un punto 3D de la nube de puntos

$$p(t) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_0 + w_{x1}t + w_{x2}t^2 \\ y_0 + w_{y1}t + w_{y2}t^2 \\ z_0 + w_{z1}t + w_{z2}t^2 \end{bmatrix}$$

$$p_0 = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}$$

El punto p_0 es el primer punto de la nube de puntos.

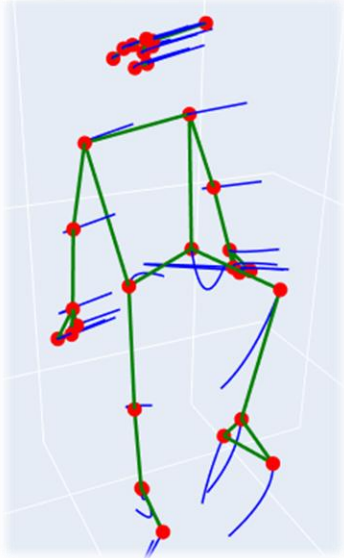
Parámetros por estimar (entrenar):

- $w_{x1}, w_{x2}, w_{y1}, w_{y2}, w_{z1}, w_{z2}$
- $t^{(1)}, \dots, t^{(m)}$

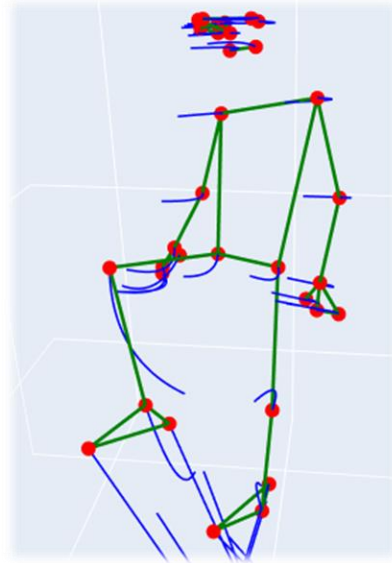
Para la parametrización de la trayectoria, se decidió utilizar polinomios de segundo orden para cada componente debido a la naturaleza de los movimientos que puede describir nuestro cuerpo partiendo de una postura y realizando el movimiento a otra postura, ya que estos por lo mucho puede ser trayectorias que se pueden curvar en un punto.

La función objetivo se minimizo utilizando el método *minimize* de la librería SciPy, en el cual se configuro como optimizador el algoritmo *BFGS* (Broyden–Fletcher–Goldfarb–Shanno algorithm).

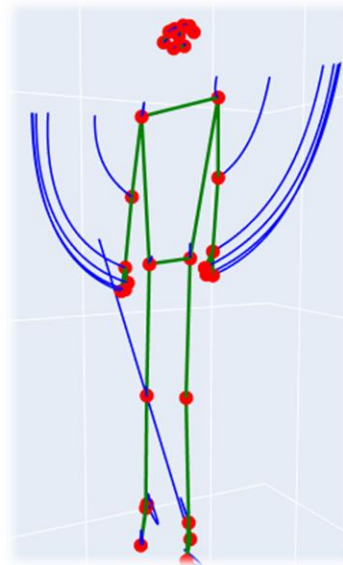
Una vez ajustada la trayectoria a una nube de puntos, lo que sigue es trazar la trayectoria, la cual describe un movimiento más suave para el punto 3D a lo largo del tiempo. Para esto se generan 100 puntos a lo largo de la trayectoria ajustada.

Nombre	Trayectoria ajustada para cada punto 3D
Levantamiento de rodilla izquierda	

Levantamiento de rodilla derecha



Levantamiento de brazos



NOTA: Para el ajuste de trayectorias, podría ser conveniente quedarnos solo con las trayectorias de algunos puntos, aquellos que si describen una trayectoria a lo largo del tiempo, ya que los puntos que por lo general están fijos suelen tener trayectorias ajustadas que dan malos resultados.

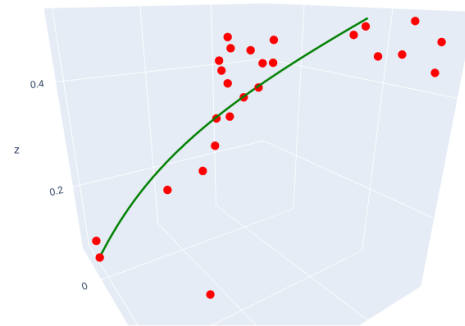
Para un punto 3D en particular, su trayectoria ajustada se vería de la siguiente manera:

Nombre	Trayectoria ajustada para cada punto 3D
--------	---

left_knee

Levantamiento de rodilla izquierda

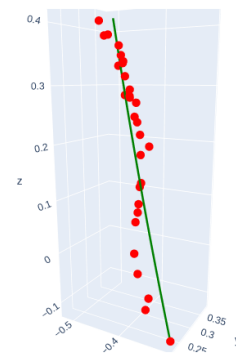
Trayectoria ajustada para la nube de puntos generada por el punto 3D “left_knee” (rodilla izquierda) a lo largo del tiempo.



right_knee

Levantamiento de rodilla derecha

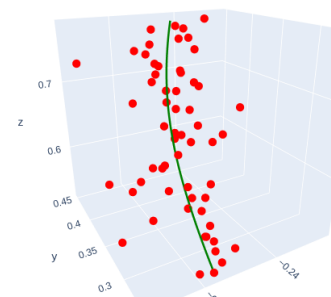
Trayectoria ajustada para la nube de puntos generada por el punto 3D “right_knee” (rodilla derecha) a lo largo del tiempo.



right_elbow

Levantamiento de brazos

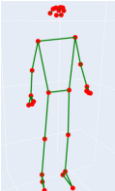
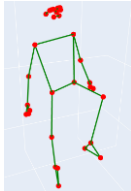
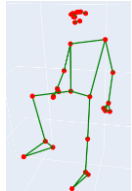
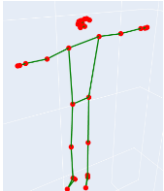
Trayectoria ajustada para la nube de puntos generada por el punto 3D “right_elbow” (codo derecho) a lo largo del tiempo.



Para obtener buenos resultados es indispensable que las estimaciones 3D de partes características del cuerpo sea buena, ya que malas estimaciones 3D solo agregan ruido, y si no es tratado ese ruido, los resultados pueden no ser los esperados.

Clasificación

Tipos de posturas a clasificar

Posturas neutras		Posturas de movimiento o ejecución	
			
De pie con los brazos a los lados	De pie levantando la rodilla izquierda	De pie levantando la rodilla derecha	De pie levantando los brazos

Para cada postura (neutra y de movimiento) se creará un clasificador binario para determinar si el paciente está realizando la postura que se le está pidiendo o no.

Hay que recordar que todo ejercicio sigue los siguientes pasos:

1. Postura neutra
2. Postura de movimiento o ejecución
3. Postura neutra

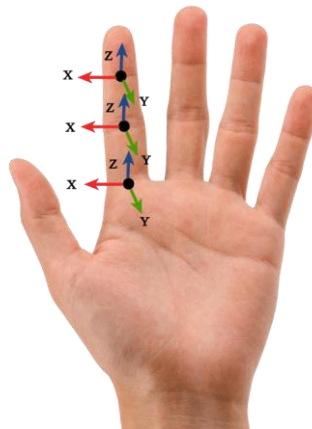
Las rutinas para cada paciente están compuestas por una serie de ejercicios con cierto número de repeticiones cada uno, por tanto, el clasificador binario que se estará usando será de acorde al ejercicio que se está pidiendo que haga el paciente, intercambiándose tres veces para clasificar postura neutra, postura de movimiento, y finalmente, postura neutra otra vez.

Obtención de los ángulos de Euler para la clasificación

Para clasificar una postura se utilizarán los ángulos de Euler asociados a matrices de rotación.

Los ángulos de Euler se calculan a partir de los puntos 3D de la postura. Cada punto 3D tiene asociado un sistema de coordenadas.

Ejemplo de sistemas de coordenadas asociados a puntos característicos del cuerpo humano



Para las estimaciones de puntos 3D característicos del cuerpo es necesario contar con modelos de deep learning que detecten esos puntos 3D pero sobre una imagen, es decir, su proyección sobre la imagen. Para esta tarea se utiliza el modelo MediaPipe Pose.

MediaPipe Pose funciona en tiempo real en dispositivos móviles e infiere 33 puntos clave del cuerpo en 2D y en 3D, y además infiere una máscara de segmentación de fondo en todo el cuerpo a partir de una imagen RGB. Cabe destacar que los puntos 3D están conformados por los puntos 2D y otro valor para cada punto que indica la profundidad relativa de dicho punto clave.

El modelo MediaPipe Pose tiene las siguientes características:

- Presenta un seguimiento de la postura del cuerpo a lo largo del tiempo
- Presenta una arquitectura de red encoder-decoder para predecir mapas de calor para todas las articulaciones presentes sobre la imagen y regresión para las coordenadas de cada punto clave
- Es muy importante que la cabeza de la persona a detectar este visible en la imagen, ya que se cuenta con un detector facial rápido que detecta primero la cabeza y después de eso predice parámetros de alineación adicionales específicos de la persona, como lo son:
 - Punto medio entre las caderas de la persona
 - Tamaño del círculo que circunda a toda la persona
 - Inclinación

Los ángulos de Euler se calculan de la siguiente manera:

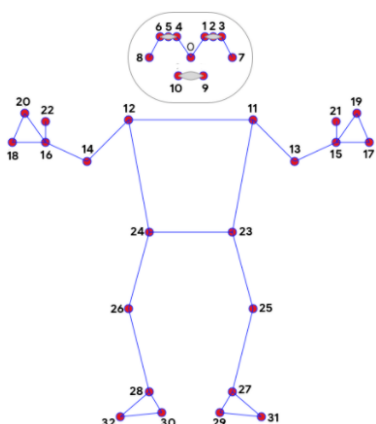
- Se eligen dos sistemas de coordenadas asociados a dos puntos 3D
- Se expresa la matriz de rotación de un sistema de coordenadas con respecto a otro
- A partir de la matriz de rotación, se calculan los ángulos de Euler asociados a la matriz de rotación

Matriz de rotación en R^3

$$R(\Phi, \theta, \Psi) = R_z(\Phi)R_y(\theta)R_x(\Psi)$$

Los ángulos de Euler están expresados en radianes.

Lo que se propuso para el cálculo de los ángulos de Euler fue lo siguiente:

Puntos de interés		Puntos característicos del cuerpo humano dados por el modelo MediaPipe Pose	
Izquierda	Derecha		
25 ← 27	26 ← 28	0. nose	17. left_pinky
23 ← 25	24 ← 26	1. left_eye_inner	18. right_pinky
11 ← 23	12 ← 24	2. left_eye	19. left_index
11 ← 13	12 ← 14	3. left_eye_outer	20. right_index
13 ← 15	14 ← 16	4. right_eye_inner	21. left_thumb
23 ← 11	24 ← 12	5. right_eye	22. right_thumb
11 ← 0	12 ← 0	6. right_eye_outer	23. left_hip
		7. left_ear	24. right_hip
		8. right_ear	25. left_knee
		9. mouth_left	26. right_knee
		10. mouth_right	27. left_ankle
		11. left_shoulder	28. right_ankle
		12. right_shoulder	29. left_heel
		13. left_elbow	30. right_heel
		14. right_elbow	31. left_foot_index
		15. left_wrist	32. right_foot_index
		16. right_wrist	

$$\mathbf{p} \leftarrow \mathbf{q}$$

Sistema de coordenadas \mathbf{q} con respecto al sistema de coordenadas \mathbf{p} (se construye \mathbf{q} a partir de \mathbf{p}).

NOTA: Todos los puntos 3D están expresados en un sistema de coordenadas global (principal).

Para una postura en particular se cuenta con:

$$(14)(3) = 42 \text{ características}$$

Dado que los ángulos de Euler están expresados en radianes, al construir el conjunto de datos no será necesario normalizar los datos.

Postura	Numero de ejemplos (instancias)
<i>Posturas aleatorias</i>	
Forman el conjunto de datos para la clase negativa	888 posturas (instancias)
<i>Postura neutra:</i>	
De pie con los brazos a los lados	224 posturas (instancias)
<i>Postura de movimiento o ejecución:</i>	
De pie levantando la rodilla izquierda	171 posturas (instancias)
<i>Postura de movimiento o ejecución:</i>	
De pie levantando la rodilla derecha	180 posturas (instancias)
<i>Postura de movimiento o ejecución:</i>	
De pie levantando los brazos	204 posturas (instancias)

Limpieza de datos

La limpieza de datos se hará de dos formas diferentes:

- Para la clase positiva, se seguirá una estrategia basada en el gráfico de boxplot
- Para la clase negativa, se seguirá una estrategia que consta en eliminar todas aquellas posturas que sean muy similares a las posturas de la clase positiva.

Para los datos de la clase positiva, es importante eliminar aquellas posturas que no pertenecen a la postura que se desea clasificar. Debido a la naturaleza de cómo se obtienen los datos, es posible que se agregue ruido a los datos.

La estrategia planteada para hacer la limpieza de los datos de la clase positiva consiste en lo siguiente:

1. Se elige una característica.
2. Se detectan aquellas posturas las cuales forman parte de los valores atípicos dado por la gráfica del boxplot.

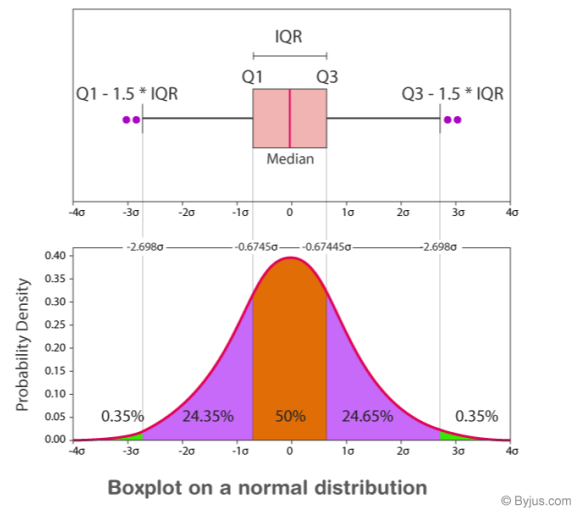
Se consideran valores atípicos a todos aquellos valores:

- Que estén por debajo de $Q1 - 1.5 IQR$
- Que estén por encima de $Q3 + 1.5 IQR$

Donde:

- Q1: Valor debajo del cual se encuentra el 25% de los datos
 - Q3: Valor debajo del cual se encuentra el 75% de los datos
 - IQR: Rango intercuartil (medida de dispersión, ya que, si IQR es grande, implica que existe mucha dispersión de los datos)
3. Se repite el proceso con todas las características, y al final se eliminan las posturas acumuladas del paso anterior.

Gráfico boxplot



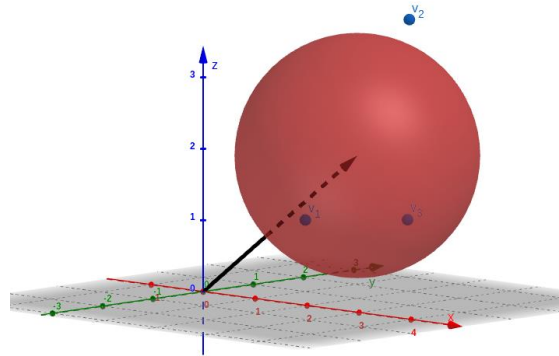
Para los datos de la clase negativa, dado que son posturas aleatorias, es posible que existan datos que sean muy similares a los de la clase positiva. Estos datos se desean eliminar de la clase negativa.

La estrategia planteada para eliminar datos de la clase negativa que son muy similares a la clase positiva consiste en lo siguiente:

- Se define el centroide de la clase positiva
- Se define una hiperesfera centrada en el centroide
- Se extiende la hiperesfera hasta cubrir un cierto porcentaje de datos de la clase positiva
 - El porcentaje de datos de la clase positiva que debe cubrir la hiperesfera es del 75 % (así se definió)
- Los datos de la clase negativa que se encuentran dentro de la hiperesfera serán los datos que no se tomarán en cuenta

El hecho de utilizar una hiperesfera es por la suposición que se hace con respecto a los datos de la clase positiva, ya que por su naturaleza se estima que la nube de puntos puede estar agrupada de forma compacta.

Ejemplo de hiperesfera en R^3 que se extiende desde el centroide de la clase positiva hasta cubrir un cierto porcentaje de datos de la clase positiva



Postura	Numero de ejemplos después de la limpieza
Postura neutra:	Clase positiva: 166
De pie con los brazos a los lados	<i>Se eliminaron 58 ejemplos de 224 (25.89 %)</i>
	Clase negativa: 888
	<i>Dada la estrategia propuesta, no se eliminó ningún ejemplo</i>
Postura de movimiento o ejecución:	Clase positiva: 76
De pie levantando la rodilla izquierda	<i>Se eliminaron 95 ejemplos de 171 (55.55 %)</i>
	Clase negativa: 888
	<i>Dada la estrategia propuesta, no se eliminó ningún ejemplo</i>
Postura de movimiento o ejecución:	Clase positiva: 99
De pie levantando la rodilla derecha	<i>Se eliminaron 81 ejemplos de 180 (45.0 %)</i>
	Clase negativa: 885
	<i>Dada la estrategia propuesta, se eliminaron 3 ejemplos de 888 (0.33 %)</i>
Postura de movimiento o ejecución:	Clase positiva: 99
De pie levantando los brazos	<i>Se eliminaron 105 ejemplos de 204 (51.47 %)</i>
	Clase negativa: 888

Dada la estrategia propuesta, no se eliminó ningún ejemplo

Creación del conjunto de datos

Una vez hecha la limpieza de datos tanto para la clase positiva como para la clase negativa, se procede a construir el conjunto de datos final.

El conjunto de datos al final tendrá m ejemplos (de m posturas) y n características.

Valores para las clases:

- Clase positiva: 1
- Clase negativa: 0

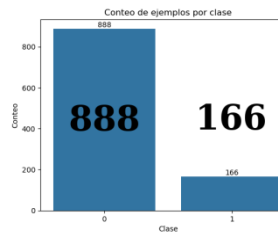
Postura	Conjunto de datos final
<i>Postura neutra:</i> De pie con los brazos a los lados	$X \in \mathbb{R}^{1054 \times 42}$ $y \in \mathbb{R}^{1054}$
<i>Postura de movimiento o ejecución:</i> De pie levantando la rodilla izquierda	$X \in \mathbb{R}^{964 \times 42}$ $y \in \mathbb{R}^{964}$
<i>Postura de movimiento o ejecución:</i> De pie levantando la rodilla derecha	$X \in \mathbb{R}^{984 \times 42}$ $y \in \mathbb{R}^{984}$
<i>Postura de movimiento o ejecución:</i> De pie levantando los brazos	$X \in \mathbb{R}^{987 \times 42}$ $y \in \mathbb{R}^{987}$

Visualización de los datos

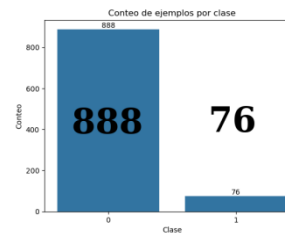
En lo general, se tendrá un desbalance enorme entre las clases, teniendo más ejemplos negativos que positivos, lo cual se debe a que existen muchas posturas aleatorias que se pueden obtener para la clase negativa.

Desbalance de clases			
<i>Postura neutra:</i> De pie con los brazos a los lados	<i>Postura de movimiento o ejecución:</i>	<i>Postura de movimiento o ejecución:</i>	<i>Postura de movimiento o ejecución:</i>

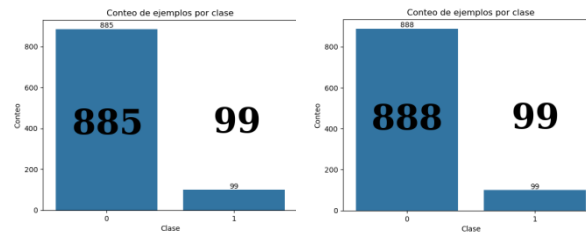
De pie levantando la rodilla izquierda



De pie levantando la rodilla derecha



De pie levantando los brazos



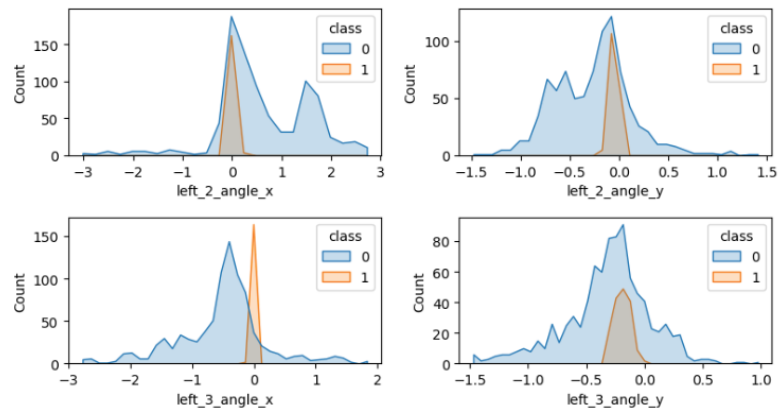
Una de las cosas que se suele hacer en un problema de clasificación cuando se tiene el conjunto de datos es visualizar la separabilidad de las clases por característica, así como se muestra a continuación (solo para algunas características):

Postura

**Visualización de la separabilidad de las clases por característica
(comportamiento de cada característica por clase)**

Postura neutra:

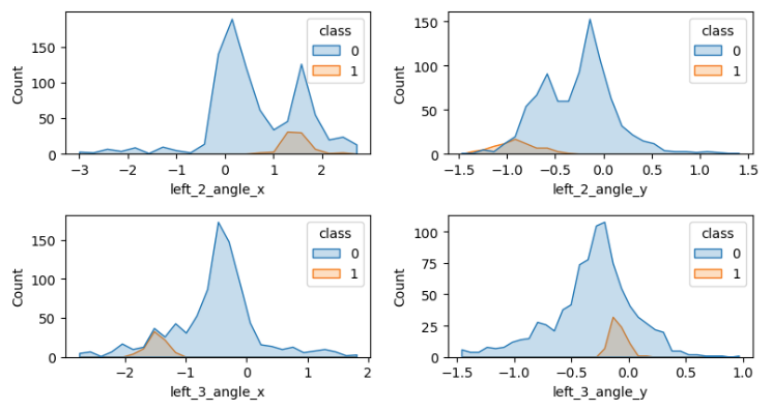
De pie con los brazos a los lados



Postura de movimiento o

ejecución:

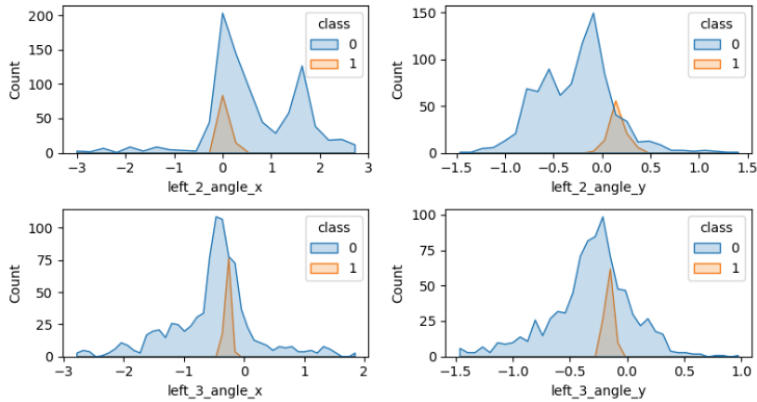
De pie levantando la rodilla izquierda



Postura de movimiento o

ejecución:

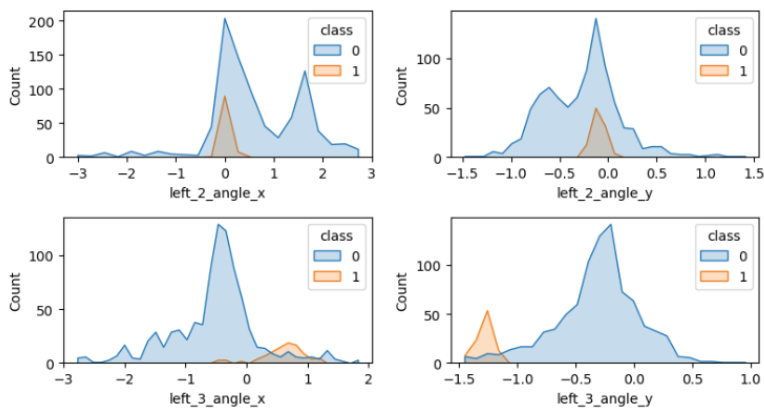
De pie levantando la rodilla derecha



Postura de movimiento o

ejecución:

De pie levantando los brazos



Es de esperarse que para una cierta postura los valores para una cierta característica de la clase positiva tiendan a estar concentrados en un pequeño rango de valores (ya que en una postura fija no hay movimiento). En cambio, para la clase negativa (que son posturas aleatorias), los valores que puede tomar una característica en concreto son muy variados.

NOTA: En el mejor de los casos, las clases tendrían que tender a separarse.

División del conjunto de datos (entrenamiento y prueba)

Para la división del conjunto de datos se utiliza la estrategia “Stratify” para que la proporción de clases sea la misma tanto en el conjunto de entrenamiento como en el conjunto de prueba.

Se optó por la estrategia 80 y 20 para la separación del conjunto de entrenamiento y prueba, respectivamente.

Conjuntos de datos obtenidos:

- Conjunto de entrenamiento (80% de los datos totales)

- Conjunto de prueba (20% de los datos totales)

Postura	Conjunto de entrenamiento	Conjunto de prueba
Postura neutra: De pie con los brazos a los lados	$X_{\text{train}} \in \mathbb{R}^{843 \times 42}$ $y_{\text{train}} \in \mathbb{R}^{843}$ Porcentaje de la clase positiva: 15 % (aprox)	$X_{\text{test}} \in \mathbb{R}^{211 \times 42}$ $y_{\text{test}} \in \mathbb{R}^{211}$ Porcentaje de la clase positiva: 15 % (aprox)
Postura de movimiento o ejecución: De pie levantando la rodilla izquierda	$X_{\text{train}} \in \mathbb{R}^{771 \times 42}$ $y_{\text{train}} \in \mathbb{R}^{771}$ Porcentaje de la clase positiva: 8 % (aprox)	$X_{\text{test}} \in \mathbb{R}^{193 \times 42}$ $y_{\text{test}} \in \mathbb{R}^{193}$ Porcentaje de la clase positiva: 8 % (aprox)
Postura de movimiento o ejecución: De pie levantando la rodilla derecha	$X_{\text{train}} \in \mathbb{R}^{787 \times 42}$ $y_{\text{train}} \in \mathbb{R}^{787}$ Porcentaje de la clase positiva: 10 % (aprox)	$X_{\text{test}} \in \mathbb{R}^{197 \times 42}$ $y_{\text{test}} \in \mathbb{R}^{197}$ Porcentaje de la clase positiva: 10 % (aprox)
Postura de movimiento o ejecución: De pie levantando los brazos	$X_{\text{train}} \in \mathbb{R}^{789 \times 42}$ $y_{\text{train}} \in \mathbb{R}^{789}$ Porcentaje de la clase positiva: 10 % (aprox)	$X_{\text{test}} \in \mathbb{R}^{198 \times 42}$ $y_{\text{test}} \in \mathbb{R}^{198}$ Porcentaje de la clase positiva: 10 % (aprox)

Selección de los mejores hiperparámetros para el modelo de clasificación binaria

Para la tarea de clasificación se utilizarán las redes neuronales profundas, esto debido a que pueden aprender límites de decisión complejos, lo cual es lo que se busca.

Para tratar el desbalance de clases, se ha decidido ponderar las clases de la siguiente manera:

Ponderación clase negativa:	1
Ponderación clase positiva:	$\frac{n_-}{n_+}$

Donde:

- n_- : Numero de datos negativos utilizados en el entrenamiento
- n_+ : Numero de datos positivos utilizados en el entrenamiento

Con respecto a las arquitecturas de la red, estas fueron las que se propusieron:

Arquitectura 1								
Capa 0 (capa de entrada)	Capa 1		Capa 2		Capa 3		Capa 4 (capa de salida)	
X	Neuronas	F.A.	Neuronas	F.A.	Neuronas	F.A.	Neuronas	F.A.
	5	Sigmoide	5	Sigmoide	5	Sigmoide	2	Softmax

Arquitectura 2						
Capa 0 (capa de entrada)	Capa 1		Capa 2		Capa 3 (capa de salida)	
X	Neuronas	F.A.	Neuronas	F.A.	Neuronas	F.A.
	30	Sigmoide	20	Sigmoide	2	Softmax

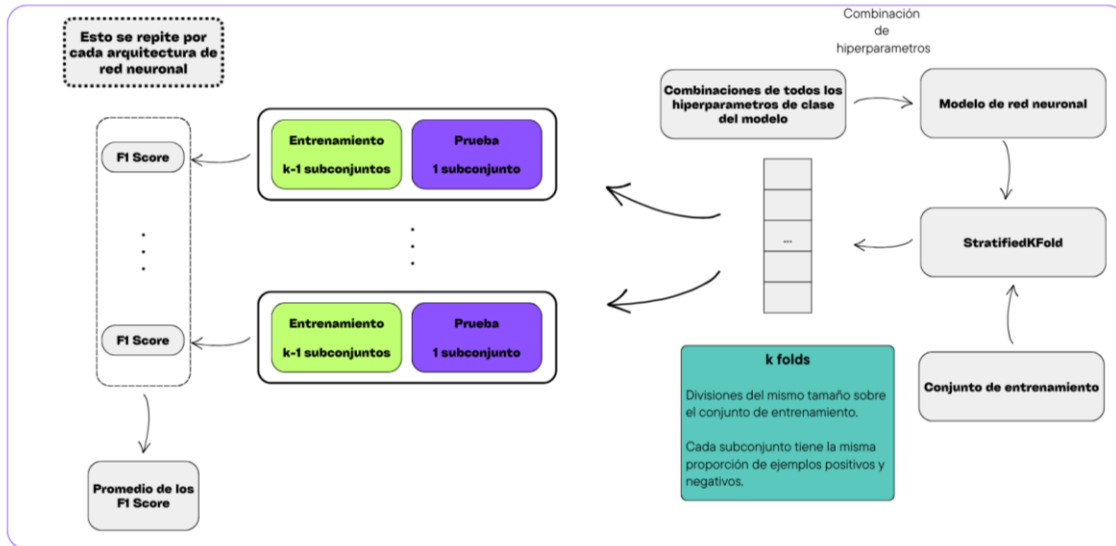
F.A.: Función de activación

El resultado de la capa de salida va hacia la función de perdida, la cual cuantifica el error entre lo predicho por el modelo y lo real.

La estrategia que se utilizó para la selección de la mejor arquitectura de red neuronal y de los mejores hiperparámetros es la de "StratifiedKFold". El número de folds utilizados fue de 5, y el número de épocas fue de 100. Además, se definió una tolerancia de $1e-6$ (0.000001) para que el entrenamiento se interrumpa cuando la función de perdida no haya tenido un cambio entre épocas mayor a la tolerancia.

El entrenamiento de las redes neuronales se llevó a cabo en una librería de machine learning que se creó con el propósito de utilizarla en todo momento durante el desarrollo del proyecto.

Flujo de entrenamiento:



Selección de la mejor arquitectura de red neuronal y de los mejores hiperparámetros:

- Para cada arquitectura, se elige la mejor combinación de hiperparámetros
 - Combinación 1: F1 Score promedio asociado
 - ...
 - Combinación p: F1 Score promedio asociado

Se elige la combinación de hiperparámetros con el F1 Score promedio más alto.

- La mejor arquitectura es aquella con el F1 Score promedio más alto

Para la creación de todas las combinaciones posibles de hiperparámetros se define un diccionario, así como se muestra a continuación:

```
class_parameters_dict_list={
  "optimizer": [
    {
      "name": "rmsprop",
      "parameters": {
        "learning_rate": [0.1, 0.01],
        "momentum": [0.5, 0.9],
      }
    },
    {
      "name": "adam",
      "parameters": {
        "learning_rate": [0.1, 0.01],
        "beta_1": [0.9],
        "beta_2": [0.999],
      }
    }
  ],
  "loss_function": [
    {
      "name": "cce",
      "parameters": {
        "weights": [weights]
      }
    }
  ]
}
```

A partir del diccionario anterior, se obtienen las siguientes combinaciones de hiperparametros:

Combinación 1	Combinación 2	Combinación 3
<pre>{ "loss_function":{ "name":"cce", "parameters":{ "weights":{ "0": "1": } } }, "optimizer":{ "name":"rmsprop", "parameters":{ "momentum":0.5, "learning_rate":0.1 } } }</pre>	<pre>{ "loss_function":{ "name":"cce", "parameters":{ "weights":{ "0": "1": } } }, "optimizer":{ "name":"rmsprop", "parameters":{ "momentum":0.5, "learning_rate":0.01 } } }</pre>	<pre>{ "loss_function":{ "name":"cce", "parameters":{ "weights":{ "0": "1": } } }, "optimizer":{ "name":"rmsprop", "parameters":{ "momentum":0.9, "learning_rate":0.1 } } }</pre>
Combinación 4	Combinación 5	Combinación 6

```
{
  "loss_function":{
    "name":"cce",
    "parameters":{
      "weights":{
        "0": 5.33,
        "1": 1.00
      }
    }
  },
  "optimizer":{
    "name":"rmsprop",
    "parameters":{
      "momentum":0.9,
      "learning_rate":0.01
    }
  }
}
```

```
{
  "loss_function":{
    "name":"cce",
    "parameters":{
      "weights":{
        "0": 11.63,
        "1": 1.00
      }
    }
  },
  "optimizer":{
    "name":"adam",
    "parameters":{
      "beta_2":0.999,
      "beta_1":0.9,
      "learning_rate":0.1
    }
  }
}
```

```
{
  "loss_function":{
    "name":"cce",
    "parameters":{
      "weights":{
        "0": 8.98,
        "1": 1.00
      }
    }
  },
  "optimizer":{
    "name":"adam",
    "parameters":{
      "beta_2":0.999,
      "beta_1":0.9,
      "learning_rate":0.01
    }
  }
}
```

Pesos asignados a las clases para los diferentes clasificadores de posturas (de acorde a la estrategia planteada)

<i>Postura neutra:</i> De pie con los brazos a los lados	<i>Postura de movimiento o ejecución:</i> De pie levantando la rodilla izquierda	<i>Postura de movimiento o ejecución:</i> De pie levantando la rodilla derecha	<i>Postura de movimiento o ejecución:</i> De pie levantando los brazos
Clase positiva: 5.33 Clase negativa: 1.00	Clase positiva: 11.63 Clase negativa: 1.00	Clase positiva: 8.96 Clase negativa: 1.00	Clase positiva: 8.98 Clase negativa: 1.00

Resultados

Clasificación

F1 Score de cada combinación de hiperparametros

Resultados obtenidos de la métrica “F1 Score” para cada arquitectura y para cada combinación de hiperparametros:

Postura		
<i>Postura neutra:</i>	Arquitectura	Métrica 'F1 Score' promedio
De pie con los brazos a los lados	1	■ Combinacion 1: 0.9868933766165906
		■ Combinacion 2: 0.9649418424674934
		■ Combinacion 3: 0.989188422085282
		■ Combinacion 4: 0.980564696238415
		■ Combinacion 5: 0.9911741528762805
		■ Combinacion 6: 0.9195740482228457
	2	■ Combinacion 1: 0.9912439990208579
		■ Combinacion 2: 0.9868959754665539
		■ Combinacion 3: 0.9934155377294992
		■ Combinacion 4: 0.9911753883862096
		■ Combinacion 5: 0.9934155377294992
		■ Combinacion 6: 0.9934155377294992
	Mejor arquitectura: 2	
	Mejor combinación de hiperparámetros: 3	
<i>Postura de movimiento o ejecución:</i>	Arquitectura	Métrica 'F1 Score' promedio
De pie levantando la rodilla izquierda	1	■ Combinacion 1: 0.963183958316425
		■ Combinacion 2: 0.8312989620131876
		■ Combinacion 3: 0.9753626850858991
		■ Combinacion 4: 0.801343989655242

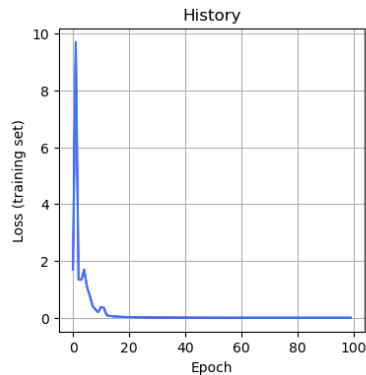
		<ul style="list-style-type: none"> ■ Combinacion 5: 0.9700878241345112 ■ Combinacion 6: 0.5175870477058216
	2	<ul style="list-style-type: none"> ■ Combinacion 1: 0.9875414118553731 ■ Combinacion 2: 0.9831880549649139 ■ Combinacion 3: 0.9875414118553731 ■ Combinacion 4: 0.9872362256249183 ■ Combinacion 5: 0.9875414118553731 ■ Combinacion 6: 0.9915895825153775
	Mejor arquitectura: 2 Mejor combinación de hiperparámetros: 6	
Postura de	Arquitectura	Métrica 'F1 Score' promedio
movimiento o	1	<ul style="list-style-type: none"> ■ Combinacion 1: 0.9930407801418439 ■ Combinacion 2: 0.9060011491672476 ■ Combinacion 3: 0.9930407801418439 ■ Combinacion 4: 0.9573701093340562 ■ Combinacion 5: 0.9862709453046575 ■ Combinacion 6: 0.9730801829893949
ejecución:	2	<ul style="list-style-type: none"> ■ Combinacion 1: 1.0 ■ Combinacion 2: 0.9930346617415037 ■ Combinacion 3: 1.0 ■ Combinacion 4: 1.0 ■ Combinacion 5: 0.9930346617415037 ■ Combinacion 6: 0.9896510018207414
	Mejor arquitectura: 2 Mejor combinación de hiperparámetros: 1	
Postura de	Arquitectura	Métrica 'F1 Score' promedio
movimiento o	1	<ul style="list-style-type: none"> ■ Combinacion 1: 0.9966163400792375 ■ Combinacion 2: 0.9932326801584752 ■ Combinacion 3: 0.9966163400792375 ■ Combinacion 4: 0.9966163400792375
ejecución:		
De pie levantando la rodilla derecha		
De pie levantando los brazos		

	<ul style="list-style-type: none"> ■ Combinacion 5: 0.9932326801584752 ■ Combinacion 6: 0.9930371767371653
2	<ul style="list-style-type: none"> ■ Combinacion 1: 1.0 ■ Combinacion 2: 1.0 ■ Combinacion 3: 1.0 ■ Combinacion 4: 0.9930371767371653 ■ Combinacion 5: 1.0 ■ Combinacion 6: 0.9966163400792375
Mejor arquitectura: 2 Mejor combinación de hiperparametros: 1	

Al final, para cada clasificador de postura:

- El mejor modelo se entrena con todo el conjunto de entrenamiento
- El mejor modelo cuenta con la mejor arquitectura y los mejores hiperparámetros
- Los parámetros de la red neuronal y los mejores hiperparámetros son guardados para su posterior uso

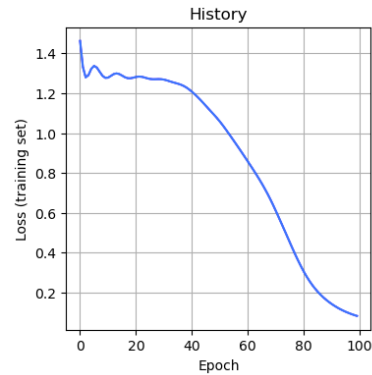
Historial del costo de la función de perdida para cada clasificador de postura (después de entrenarlo con todo el conjunto de entrenamiento):

Postura	Costo de la función de perdida en cada época
Postura neutra: De pie con los brazos a los lados	

Postura de movimiento o

ejecución:

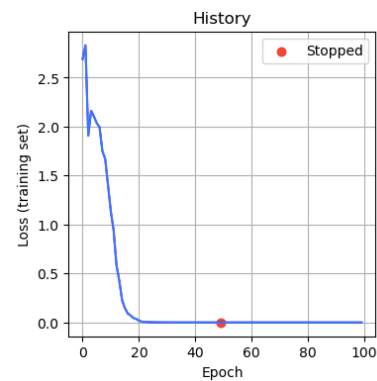
De pie levantando la rodilla
izquierda



Postura de movimiento o

ejecución:

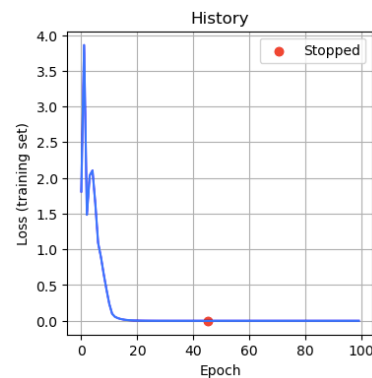
De pie levantando la rodilla
derecha



Postura de movimiento o

ejecución:

De pie levantando los brazos



Resultados con el conjunto de prueba

Para medir el desempeño del modelo se utilizó la métrica “F1 Score”, la cual hace uso de otras métricas resultantes de la matriz de confusión.

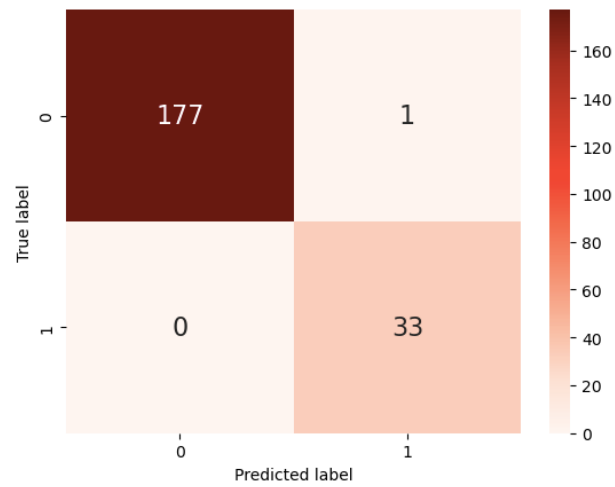
Los resultados obtenidos en el conjunto de prueba fueron los siguientes:

Postura	Matriz de confusión
---------	---------------------

Postura neutra:

De pie con los brazos a los
lados

F1 Score: 0.9911288627286104

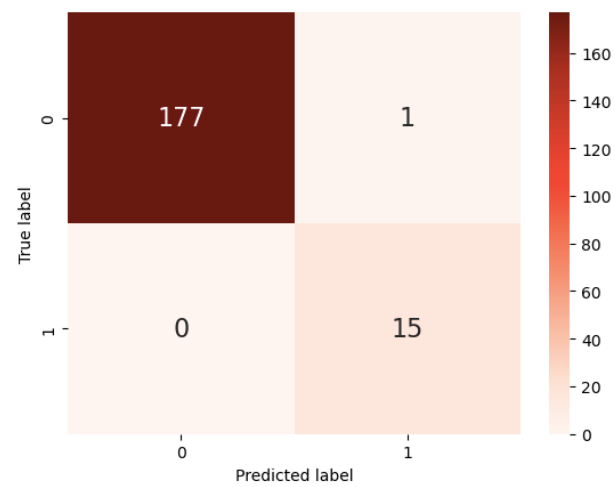


Postura de movimiento o

ejecución:

De pie levantando la rodilla
izquierda

F1 Score: 0.9824625170377101

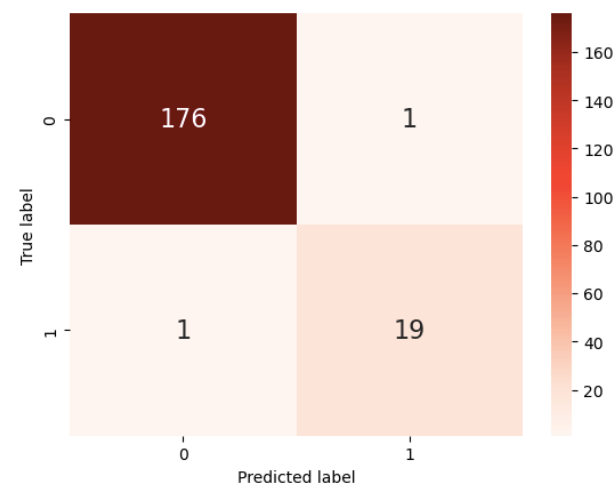


Postura de movimiento o

ejecución:

De pie levantando la rodilla
derecha

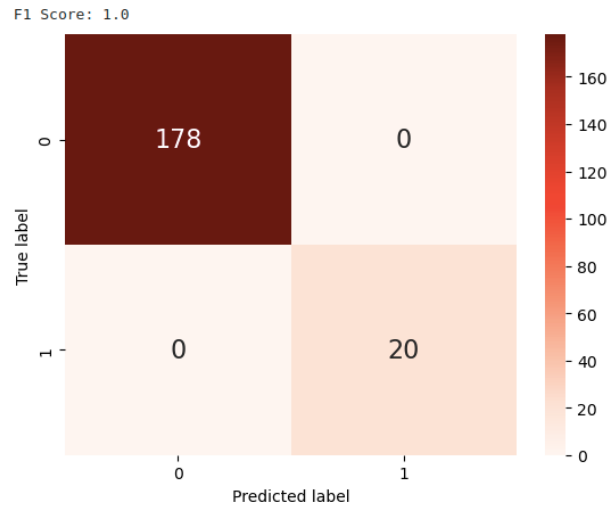
F1 Score: 0.9721751412429379



Postura de movimiento o

ejecución:

De pie levantando los brazos



Métricas de evaluación

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

	Predicted Positive	Predicted Negative	
Actual Positive	TP <i>True Positive</i>	FN <i>False Negative</i>	Sensitivity $\frac{TP}{(TP + FN)}$
Actual Negative	FP <i>False Positive</i>	TN <i>True Negative</i>	Specificity $\frac{TN}{(TN + FP)}$
	Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

Interpretación de las métricas:

- Sensitivity (Recall)

- De todos los ejemplos positivos, ¿qué tanto acertó el modelo?

- Specificity

- De todos los ejemplos negativos, ¿qué tanto acertó el modelo?

- Precision

-
- De todas las predicciones positivas, ¿qué tanto acertó el modelo?
 - *Accuracy*
 - Predicciones correctas / Total
 - *F1 Score*
 - Es una métrica robusta que toma las métricas 'Precision' y 'Recall'.
 - Devuelve cercano a 1 si ambas son también cercanas a 1
 - Si una es más grande que la otra, devuelve valores bajos
 - Si ambas son bajas, devuelve un valor cercano a 0
-

Regresión

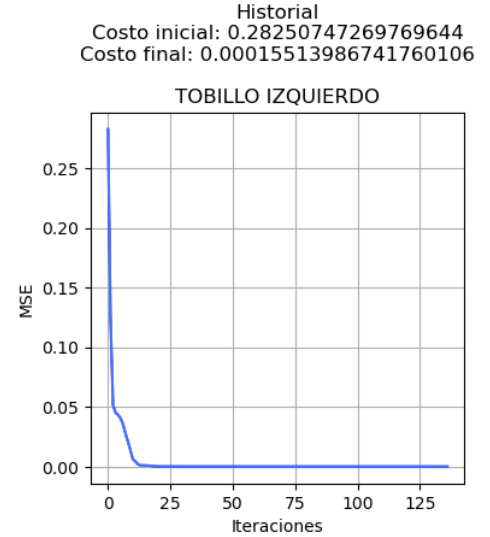
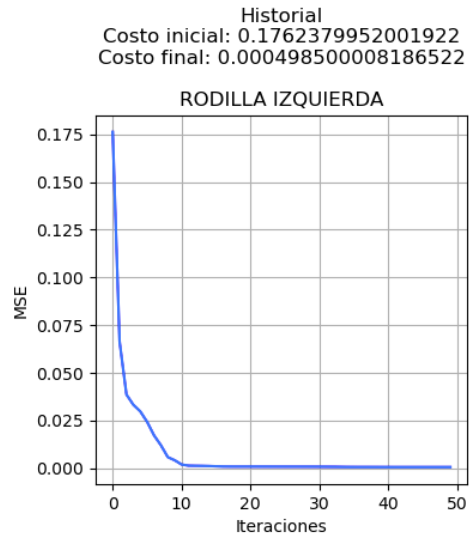
Debido a la gran cantidad de trayectorias que se ajustan (33 trayectorias, una para cada punto 3D), solo se mostraran las trayectorias más relevantes y los resultados obtenidos.

Para medir el desempeño de las trayectorias ajustadas se utiliza la métrica MSE (error cuadrático medio).

Dado que los puntos están expresados en un sistema de coordenadas global y sus unidades se encuentran en metros, la interpretación que se le pueda dar al resultado de MSE es la distancia promedio más cercana, en metros cuadrados, que hay entre la trayectoria ajustada y cualquier otro punto de la nube de puntos.

Nombre de la trayectoria	Métrica MSE de la trayectoria ajustada	
	Rodilla izquierda	Tobillo izquierdo

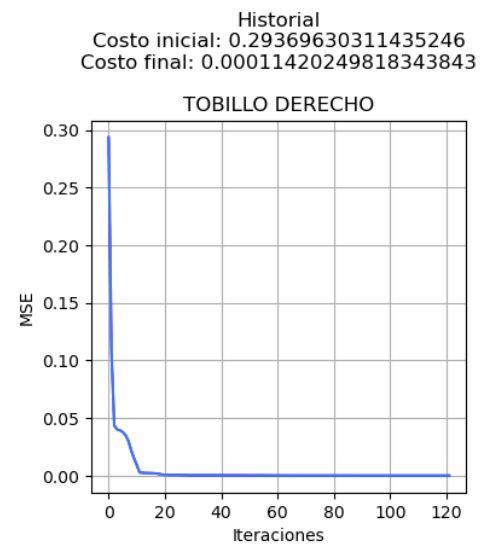
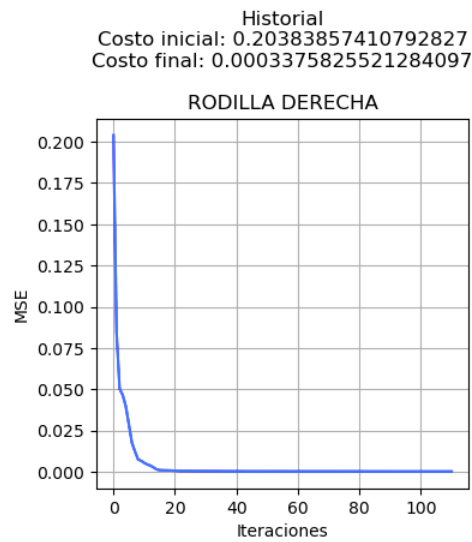
Levantamiento de rodilla izquierda



Rodilla derecha

Tobillo derecho

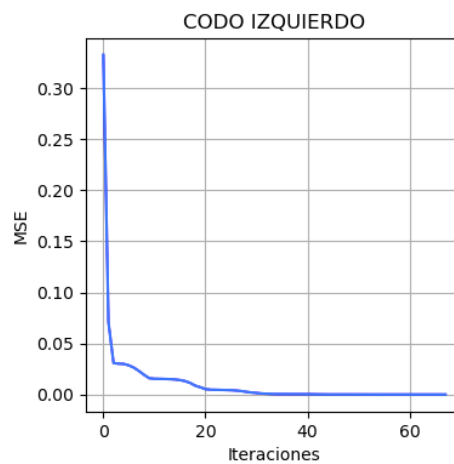
Levantamiento de rodilla derecha



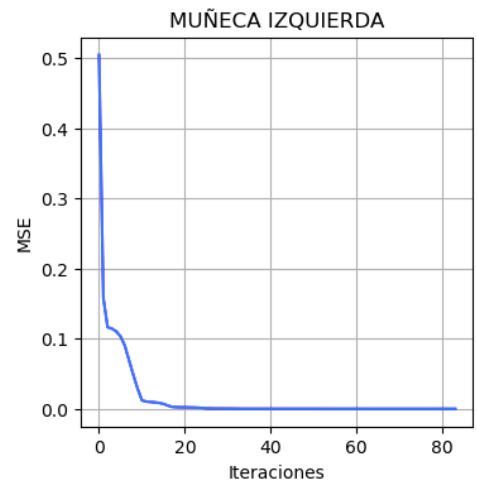
Codo izquierdo

Muñeca izquierda

Historial
Costo inicial: 0.3323512708621915
Costo final: 5.83168481036908e-05



Historial
Costo inicial: 0.5040430251885019
Costo final: 0.000137797380830982

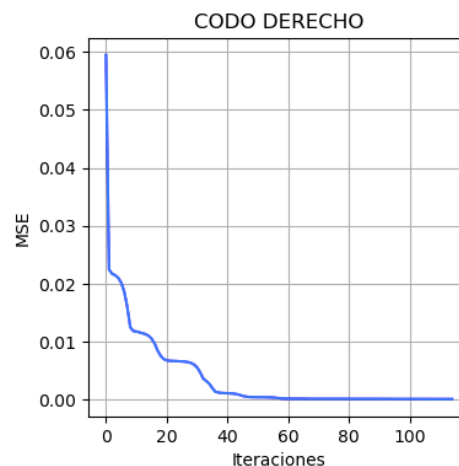


Levantamiento de
brazos

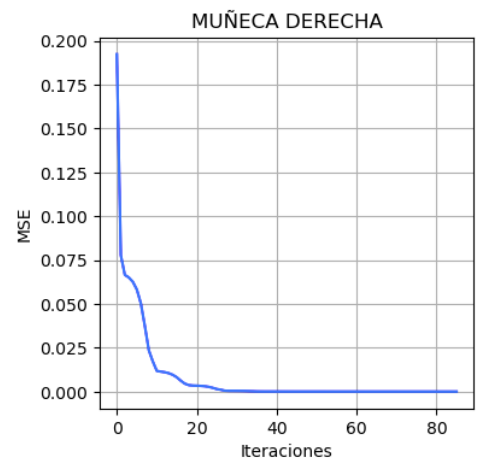
Codo derecho

Muñeca derecha

Historial
Costo inicial: 0.05946624055215919
Costo final: 9.089686180750072e-05



Historial
Costo inicial: 0.19241290424852295
Costo final: 0.00011567924077243332



Sistema informático

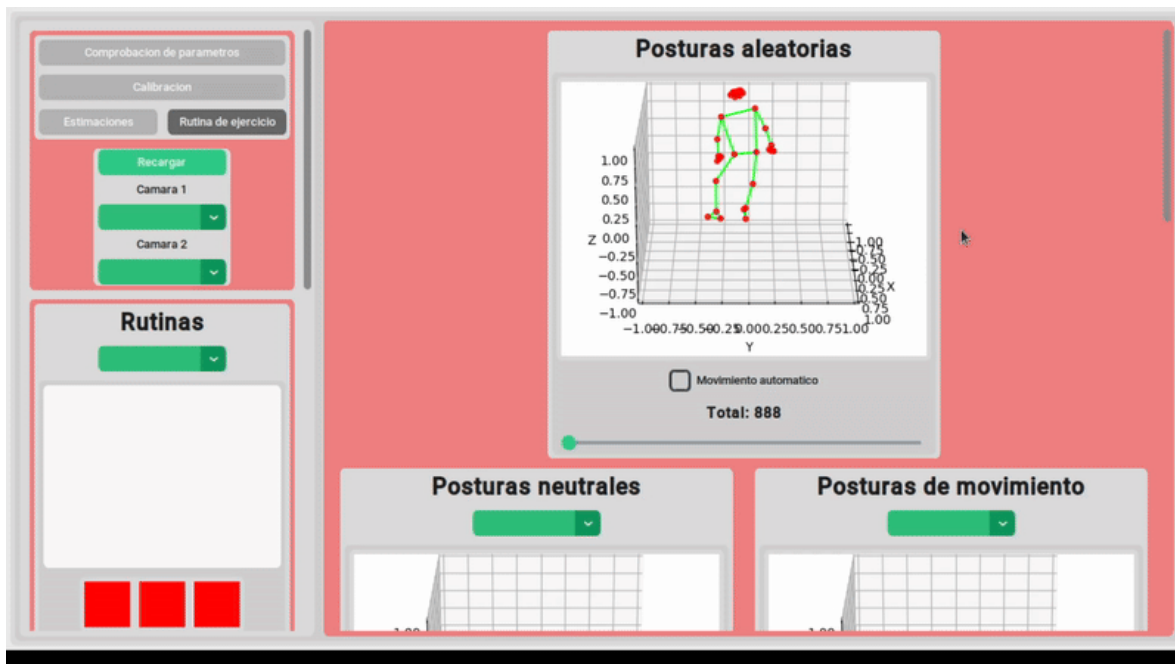
Todos los resultados obtenidos anteriormente se han integrado en un sistema informático para la rehabilitación física mediante rutinas de ejercicio que debe seguir el paciente, cuyas posturas son clasificadas para detectar cuando el paciente ha realizado un ejercicio correctamente. Además, se muestra el movimiento del ejercicio que se le está pidiendo que realice al paciente (esto por medio de las trayectorias que se ajustaron para poder mostrar una trayectoria más suave del movimiento).

Opciones del menú:

- Calibración de la cámara
- Comprobación de los parámetros de la cámara
- Estimaciones 3D
- Rutina de ejercicio y datos recopilados
- Selección de cámaras para trabajar
- Selección de rutina



Apartado para consultar todos los datos recopilados y para crear rutinas de ejercicios:



Apartado para clasificar las posturas del paciente a lo largo de cada ejercicio de una rutina especifica:



Discusión y conclusiones

Regresión

El ajuste de trayectorias de puntos 3D a lo largo del tiempo dio buenos resultados, tanto la elección de la parametrización de la trayectoria utilizando polinomios de grado 2 para cada componente, como la función de pérdida utilizada (MSE).

Es importante mencionar que los buenos resultados se obtuvieron de aquellos puntos 3D que su nube de puntos describe una trayectoria, ya que los que presentaron malos resultados fueron aquellos puntos 3D que permanecían aproximadamente en el mismo lugar a lo largo del tiempo.

Los resultados podrían mejorarse si solo se toman en cuenta para ajustar una trayectoria aquellos puntos 3D que describen una trayectoria a lo largo del tiempo, y el centroide de aquellos puntos 3D que permanecen en un solo lugar a lo largo del tiempo (ya que su nube de puntos se concentra en una esfera).

Clasificación

Los resultados obtenidos de la clasificación de las diferentes posturas (postura neutral y postura de movimiento) fue bastante bueno, obteniendo resultados de la métrica F1 Score promedio de 0.98.

Al poner a prueba dichos modelos de clasificación en tiempo real en un sistema informático, se observaron buenas clasificaciones, sin embargo, también hay posturas que son mal clasificadas, lo cual mayormente se debe a que hubo muchas posturas que no se lograron capturar para los datos de la clase negativa, por tanto, existe la posibilidad de que algunas posturas puedan ser clasificadas como positivas cuando realmente no pertenecen a dicha clase.

Planteando soluciones a los problemas que se obtuvieron, capturar más datos de posturas más diversas puede ayudar a tener todavía mejores resultados cuando se prueban los modelos en tiempo real, así también, puede ayudar probar con otros modelos de clasificación que puedan ajustarse y capturar la distribución de la clase positiva (el que un modelo se ajuste un poco a los datos de la clase positiva no es un problema, al contrario, es deseable para nuestro caso).

Finalmente, a pesar de que los modelos de clasificación (redes neuronales profundas) fueron entrenadas en una librería de machine learning que se creó como parte del proyecto,

las predicciones en tiempo real no fueron un problema de rendimiento, ya que solo se realiza el “forward propagation” (construcción del grafo computacional) para obtener las predicciones, no hay cálculo de derivadas en el proceso (ya que no son necesarias, solo en el entrenamiento).

Conclusión general

Los resultados pueden mejorarse bastante si se mejoran todavía más los procesos de optimización que ocurren en la visión estéreo, tanto la calibración de las cámaras (el problema de optimización que se resuelve, los parámetros de inicialización y el algoritmo de optimización), la corrección de la distorsión de la cámara y el cálculo de puntos 3D a partir de los puntos 2D correspondientes en cada cámara. Esto traería mejores resultados en lo que respecta a la estimación de la pose en 3D.

Dejando de lado la visión estéreo, otras mejoras son el encontrar mejores algoritmos de detección de puntos 2D característicos del cuerpo humano sobre una imagen (que sean mejores a MediaPipe Pose), ya que la estimación de puntos 3D requiere sí o sí que los puntos 2D correspondientes sean precisos.

El uso de sensores, en conjunto con la visión estéreo, podría dar precisión al momento de estimar una pose 3D, ya que se estarían combinando diferentes fuentes de información, lo cual conllevaría a tener mejores resultados al momento de clasificar una postura de una persona.

Finalmente, la implementación de otro modelo de clasificación podría ayudar a tener mejores resultados, tal podría ser el caso de implementar un modelo de hipersfera que pueda ser moldeable a los datos de la clase positiva, ya que como tal esto agregaría robustez y estaría clasificando como positivo a todos aquellos datos que realmente están dentro de la nube de puntos de la clase positiva.

Referencias

- Keller, A. V., Torres-Espin, A., Peterson, T. A., Booker, J., O'Neill, C., Lotz, J. C., Bailey, J. F., Ferguson, A. R., & Matthew, R. P. (2022). Unsupervised Machine Learning on Motion Capture Data Uncovers Movement Strategies in Low Back Pain. *Frontiers in Bioengineering and Biotechnology*, 10, 868684. <https://doi.org/10.3389/fbioe.2022.868684>
- Liao, Y., Vakanski, A., & Xian, M. (2020). A Deep Learning Framework for Assessing Physical Rehabilitation Exercises. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(2), 468–477. <https://doi.org/10.1109/TNSRE.2020.2966249>
- Muñoz-Cardona, J. E., Henao-Gallo, O. A., & López-Herrera, J. F. (2013). Sistema de Rehabilitación basado en el Uso de Análisis Biomecánico y Videojuegos mediante el Sensor Kinect. *Tecno Lógicas, Edición Especial*, 43-54.
- Obukhov, A., Volkov, A., Pchelintsev, A., Nazarova, A., Teselkin, D., Surkova, E., & Fedorchuk, I. (2023). Examination of the Accuracy of Movement Tracking Systems for Monitoring Exercise for Musculoskeletal Rehabilitation. *Sensors*, 23(19), 8058. <https://doi.org/10.3390/s23198058>
- Phan, T. C., Pranata, A., Farragher, J., Bryant, A., Nguyen, H. T., & Chai, R. (2024). Regression-Based Machine Learning for Predicting Lifting Movement Pattern Change in People with Low Back Pain. *Sensors*, 24(4), 1337. <https://doi.org/10.3390/s24041337>
- Uhlrich, S. D., Falisse, A., Kidziński, Ł., Muccini, J., Ko, M., Chaudhari, A. S., Hicks, J. L., & Delp, S. L. (2022). OpenCap: 3D human movement dynamics from smartphone videos. *bioRxiv*. <https://doi.org/10.1101/2022.07.07.499061>
- Zago, M., Luzzago, M., Marangoni, T., De Cecco, M., Tarabini, M., & Galli, M. (2020). 3D Tracking of Human Motion Using Visual Skeletonization and Stereoscopic Vision. *Frontiers in Bioengineering and Biotechnology*, 8, 181. <https://doi.org/10.3389/fbioe.2020.00181>