Rastreo de movimientos humanos, poses e inclinaciones con modelos de clasificación

Juan Camilo Tobar Morales
Isabella Hernández Mosquera
Karen Valeria Jurado Calpa

***Abstract*—This project proposes the development of an intelligent real-time video annotation and analysis system for the automatic identification of specific human activities such as walking, turning, sitting, and standing up. The solution employs computer vision and supervised learning techniques, integrating tools like MediaPipe for tracking key joint points and classification models to detect movement patterns. Based on a database built through video capture and annotation, kinematic features such as joint velocities, angles, and trunk inclinations were extracted.**

***Resumen*–Este proyecto propone el desarrollo de un sistema inteligente de anotación y análisis de video en tiempo real para identificar automáticamente actividades humanas específicas como caminar, girar, sentarse y levantarse. Utiliza técnicas de visión por computador y aprendizaje supervisado, integrando herramientas como MediaPipe para el seguimiento de articulaciones clave y modelos de clasificación para detectar patrones de movimiento. Se construyó una base de datos mediante captura y anotación de videos, a partir de la cual se extrajeron características cinemáticas como velocidades articulares, ángulos e inclinaciones del tronco.**

Introducción

El reconocimiento automático de actividades humanas tiene aplicaciones clave en áreas como la salud, el deporte y la vigilancia. Este proyecto desarrolla un sistema inteligente capaz de identificar en tiempo real acciones como caminar, girar, sentarse y levantarse, utilizando visión por computador y aprendizaje supervisado. Se emplearon herramientas como MediaPipe para el seguimiento articular y modelos de clasificación para analizar patrones de movimiento. Esta solución busca facilitar el análisis biomecánico y apoyar procesos de rehabilitación y monitoreo físico.

Se aplicaron procesos de normalización para asegurar que los datos fueran consistentes y comparables entre distintos individuos. Para la clasificación, se utilizaron enfoques supervisados adecuados para identificar patrones en datos complejos. Asimismo, se ajustaron ciertos parámetros del modelo con el fin de mejorar su precisión y garantizar un buen desempeño durante las predicciones.

Marco teorico

Exploración y limpieza de datos.
Es el proceso inicial donde se revisan los datos para detectar errores, valores faltantes o inconsistencias. La limpieza asegura que la información esté en buen estado antes de entrenar los modelos.

Datos atípicos.
Son valores que se alejan del comportamiento normal del conjunto de datos. Es importante detectarlos y tratarlos para evitar que afecten negativamente el rendimiento del modelo.

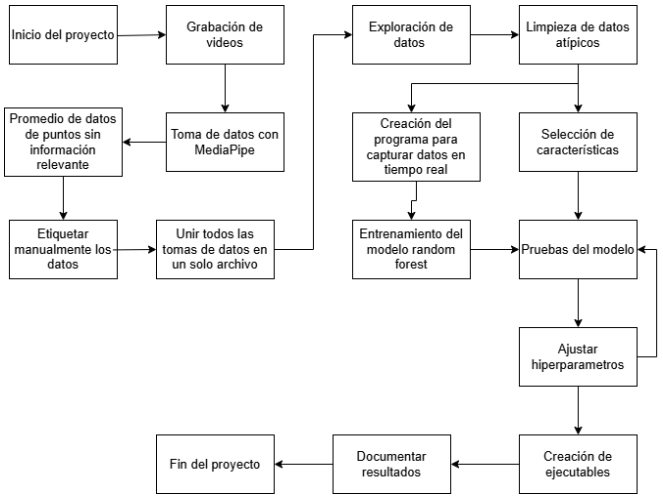
Modelo de clasificación.
Es un algoritmo que aprende a identificar categorías o clases a partir de ejemplos previos. Se utiliza para predecir la actividad que está realizando una persona según sus movimientos.

Movimientos y poses articulares.
Se refiere a las posiciones y desplazamientos de las articulaciones del cuerpo. Estas poses permiten describir acciones como caminar o sentarse, y se usan como base para el análisis del movimiento.

Random Forest.
Es un modelo de clasificación que combina varios árboles de decisión. Funciona bien con datos variados y ayuda a mejorar la precisión evitando errores comunes como el sobreajuste.

Metodologia

Se presenta el flujo de actividades que se realizó para la realización del proyecto:



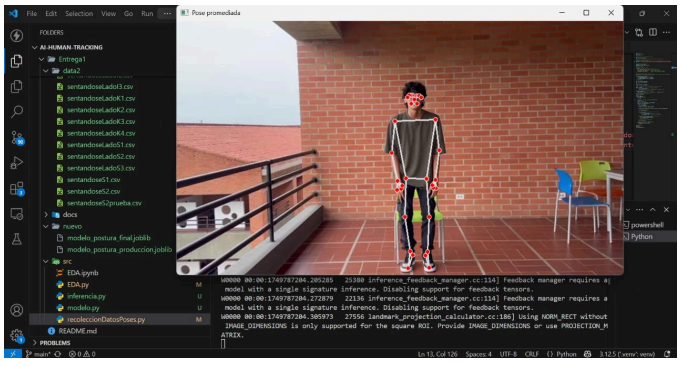
Recolección de datos.

Toma de datos: Para el modelo se registraron 57 videos de aproximadamente 10 segundos, dentro de cada video se hacían gestos diferentes, como sentarse, inclinarse, girar, pararse o caminar. Se utilizaron a 4 personas diferentes con el propósito de que haya más variabilidad.

Extracción y promedio de características irrelevantes: Con cada video se extrajo 54 landmarks brutos y 15 landmarks que representan el promedio de lugares con información irrelevante, como los diferentes puntos de la cara, los diferentes puntos de las manos y los diferentes puntos de los pies. Estos puntos fueron promediados y puestos en un solo punto.

Etiquetar datos: Luego de la toma, se etiquetaron los datos manualmente viendo desde que frame hasta que frame del video sucede que movimiento o posición. Se puso sumo cuidado para garantizar que el proceso de etiquetado fue adecuado y no afecte los datos o el proceso de aprendizaje del modelo en el futuro.

Con esto en mente se muestra un poco de como fue el proceso de recolección de los datos con el código desarrollado:



Procesamiento de datos.

Gracias al código empleado para la extracción de datos de los landmarks, se pudieron obtener los datos y ponerlos en hojas de excel. Los landmarks extraídos fueron todos los dispuestos en el cuerpo pero promediando los puntos de la mano, los puntos de los pies y los puntos de la cara. Los datos fueron dispuestos de esta manera:

A	B	C	D	E	F	G
segundo	landmark_10_	landmark_10_	landmark_10_	landmark_11_	landmark_11_	landmark
2.85	0.489824682	-0.041247755	-0.328458398	0.546513676	0.063066959	-0.25151
2.88	0.486441195	-0.042774923	-0.378111034	0.550609707	0.063253469	-0.21395
2.92	0.486839354	-0.030136091	-0.402728796	0.550281941	0.063636027	-0.23583
2.95	0.487542152	-0.029592500	-0.385952591	0.548677802	0.063829280	-0.24440
2.98	0.486611753	-0.030031563	-0.404530853	0.548264205	0.064743183	-0.23297
3.02	0.485901266	-0.028089635	-0.426074206	0.546331763	0.067082040	-0.23376
3.05	0.485629571	-0.026980500	-0.413111060	0.542357087	0.069399513	-0.23136
3.08	0.483786404	-0.027432724	-0.412848651	0.541716396	0.071868613	-0.22413
3.12	0.483065605	-0.026449935	-0.424538135	0.539293169	0.072877183	-0.23242
3.15	0.478695005	-0.021751740	-0.421445190	0.537389695	0.073183916	-0.22835
3.18	0.478398323	-0.015899609	-0.413380295	0.536446094	0.074123680	-0.22384
3.22	0.474412858	-0.014133537	-0.395255506	0.536322236	0.076024234	-0.19501
3.25	0.469259917	-0.015076134	-0.383743226	0.535740911	0.077663019	-0.18812

Posterior a esto, se realizó un análisis exploratorio de datos:

EDA.

Antes de entrenar el modelo, se realizó un análisis exploratorio de todos los datos que fueron extraídos de los videos. Esto con el propósito de poder manejar ciertos sesgos en los datos, entonces se realizó lo siguiente:

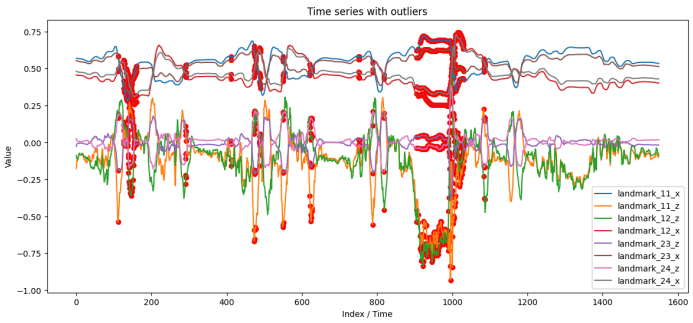
- **Tipos de datos:** Se visualizó que en qué tipo de dato se guardaron los landmarks de los videos:

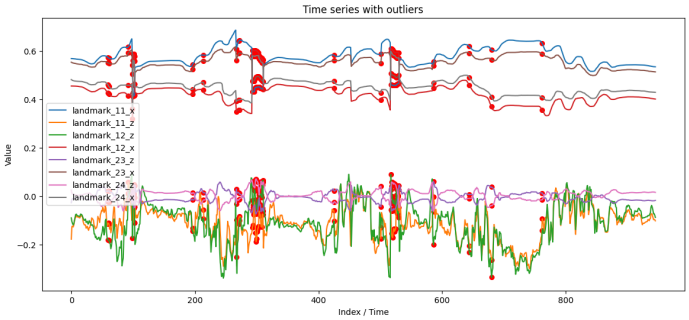
```
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   segundo         112 non-null    float64
 1   landmark_10_x   112 non-null    float64
 2   landmark_10_y   112 non-null    float64
 3   landmark_10_z   112 non-null    float64
 4   landmark_11_x   112 non-null    float64
 5   landmark_11_y   112 non-null    float64
 6   landmark_11_z   112 non-null    float64
 7   landmark_12_x   112 non-null    float64
 8   landmark_12_y   112 non-null    float64
 9   landmark_12_z   112 non-null    float64
 10  landmark_13_x   112 non-null    float64
 11  landmark_13_y   112 non-null    float64
 12  landmark_13_z   112 non-null    float64
 13  landmark_14_x   112 non-null    float64
 14  landmark_14_y   112 non-null    float64
 15  landmark_14_z   112 non-null    float64
 16  landmark_23_x   112 non-null    float64
 17  landmark_23_y   112 non-null    float64
 18  landmark_23_z   112 non-null    object
 19  landmark_24_x   112 non-null    float64
 20  landmark_24_y   112 non-null    float64
 21  landmark_24_z   112 non-null    object
 22  landmark_25_x   112 non-null    float64
```

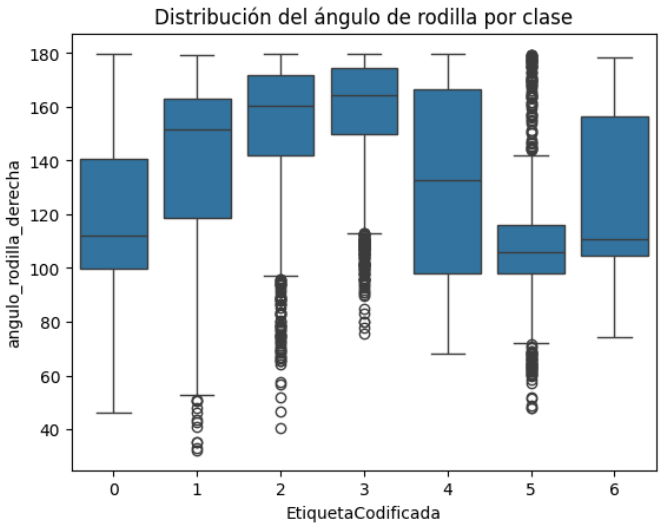
Los datos que no son procesables fácilmente como los objetos fueron cambiados a float64 con el fin de poder manejarlos como una variable numérica.

- **Validación de datos nulos:** Se visualizaron todos los conjuntos de datos por datos nulos. Los datos nulos fueron imputados con el promedio de la columna.
- **Análisis de datos atípicos:** Para poder evitar sesgos y variabilidad no necesaria, entonces, evaluamos que

tantos datos atípicos hay en los datos recolectados, se utilizaron gráfica de líneas y gráficos boxplot, como a continuación se muestra:







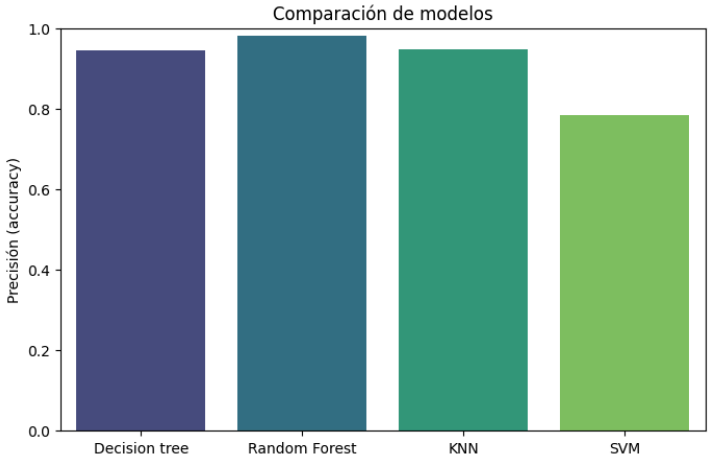
Como se puede observar existen datos atípicos por lo que se tienen que tratar para evitar variabilidad no deseada.

- **Limpieza de datos atípicos:** Se eliminaron todos los valores atípicos que pasan el umbral z-score de 2.8.

Selección de modelos a entrenar.

Se quiso saber que modelo era más adecuado para el problema abordado, con eso en mente se probaron los modelos Desicion tree, Random forest, KNN y SVM. Con esto en mente los resultados de precisión fueron los siguientes:

- Desicion tree: 0,946
- Random forest: 0,981
- KNN: 0,949
- SVM: 0,786



A partir de los resultados se tomó la decisión de continuar el desarrollo del proyecto con un modelo Random Forest.

Entrenamiento de modelo.

- **División del conjunto de datos:** Se dividió el conjunto en datos de entrenamiento y prueba con una proporción del 80% para entrenamiento y 20% para prueba. Se utilizó stratify=y para preservar la distribución de clases.
- **Ajuste de Hiperparámetros con GridSearchCV:** Se utilizó un RandomForestClassifier como estimador base, con random_state=42 para asegurar la reproducibilidad.
- **Ejecución de GridSearchCV:** Se aplicó validación cruzada con 3 particiones, utilizando un solo núcleo para evaluar todas las combinaciones posibles de la grilla.
- **Selección del mejor modelo:** Se identificaron los hiperparámetros que produjeron el mejor rendimiento, y se extrajo el modelo óptimo para la evaluación final.

  Mejores parámetros: {'max_depth': 20, 'min_samples_leaf': 5, 'min_samples_split': 10, 'n_estimators': 150}

- **Predicción sobre datos de prueba:** Se utilizó el modelo optimizado para predecir las clases del conjunto de prueba.

Resultados.

Durante las primeras pruebas del modelo, se observó un claro caso de overfitting, el clasificador mostraba un desempeño muy alto en el conjunto de entrenamiento, pero no lograba generalizar correctamente frente a datos reales o no vistos.

Esta diferencia evidenció la necesidad de ajustar los hiperparámetros del modelo para mejorar su capacidad de generalización.
Con base en esto, se realizó un segundo entrenamiento utilizando una búsqueda en malla (GridSearchCV) para encontrar la combinación óptima de parámetros, tales como la profundidad máxima del árbol, el número de estimadores y los tamaños mínimos de división y hojas. Gracias a esta optimización, el modelo logró un mejor equilibrio entre precisión en entrenamiento y prueba, reduciendo significativamente el sobreajuste observado inicialmente.
A continuación, se presentan los resultados obtenidos tras este ajuste, junto con las métricas clave que evidencian la mejora en el rendimiento del modelo.

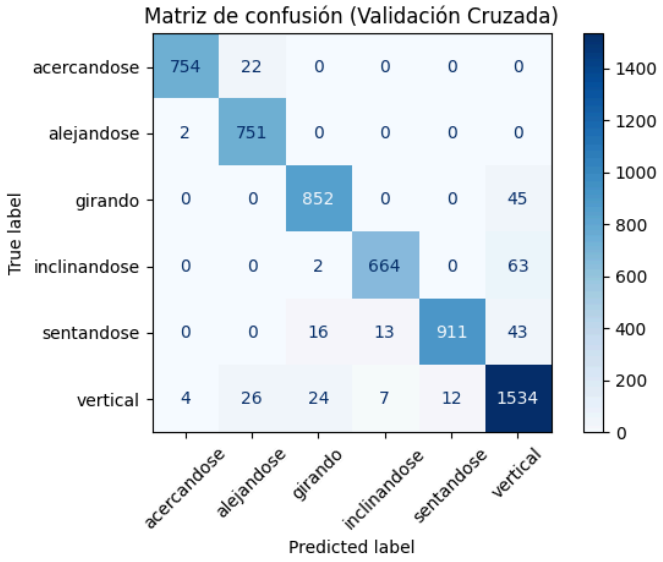
```
--- Reporte de Clasificación ---
              precision    recall  f1-score   support

  acercandose       1.00      0.99      0.99       155
   alejandose       0.99      1.00      0.99       151
     girando        0.96      0.97      0.96       179
inclinandose      0.96      0.96      0.96       146
   sentandose       0.98      0.96      0.97       197
     vertical       0.95      0.96      0.95       321

     accuracy                           0.97      1149
    macro avg       0.97      0.97      0.97      1149
 weighted avg       0.97      0.97      0.97      1149
```

Vemos que el modelo entrenado tiene una buena precisión para predecir qué movimiento se está realizando. Con esto entonces, se midieron unas métricas más generales de cómo se desempeñó en los de entrenamiento y prueba:

```
--- Diagnóstico Final ---
Precisión en entrenamiento: 0.9911
Precisión en prueba:        0.9695
```

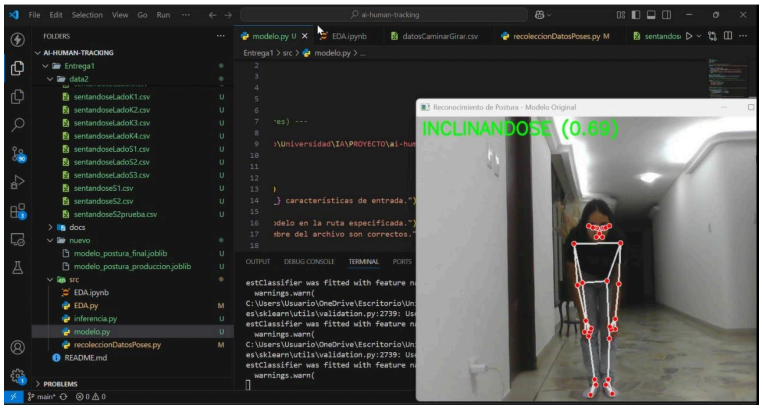
Observamos que en general el modelo es capaz de predecir adecuadamente los datos. A continuación también podemos observar la matriz de confusión y como se comporto el modelo para datos nuevos:

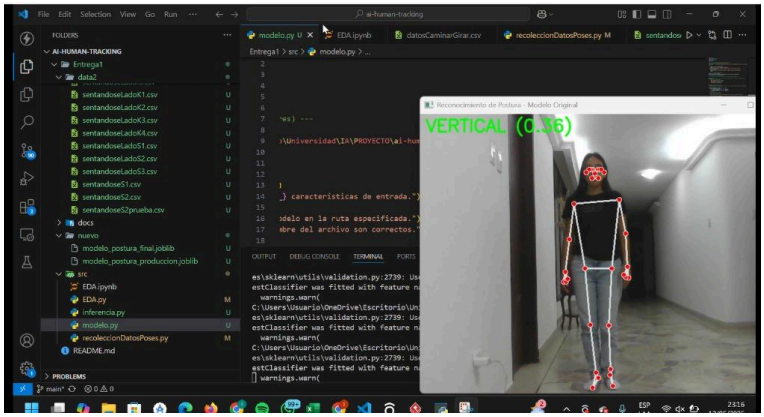
	precision	recall	f1-score	support
acercandose	0.99	0.97	0.98	776
alejandose	0.94	1.00	0.97	753
girando	0.95	0.95	0.95	897
inclinandose	0.97	0.91	0.94	729
sentandose	0.99	0.93	0.96	983
vertical	0.91	0.95	0.93	1607
accuracy			0.95	5745
macro avg	0.96	0.95	0.95	5745
weighted avg	0.95	0.95	0.95	5745

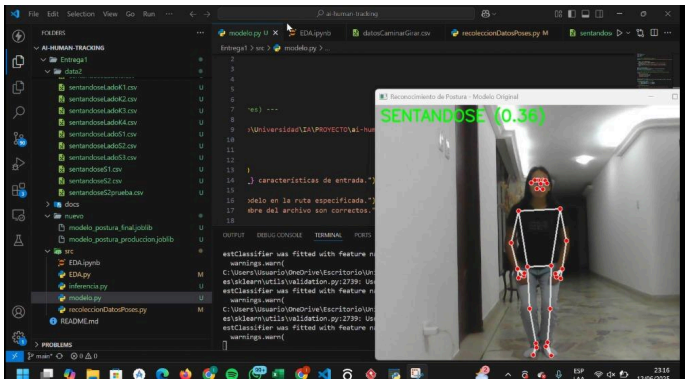


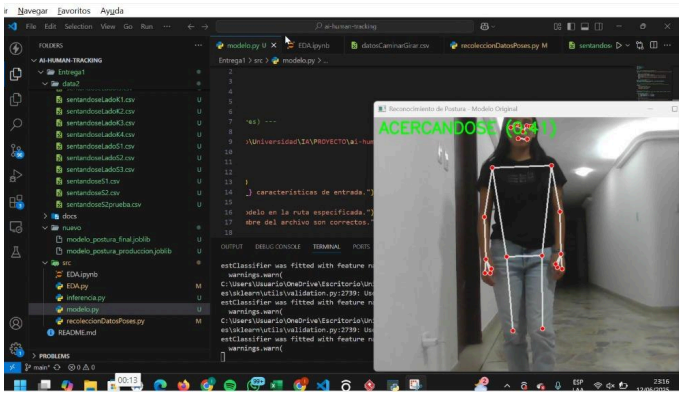
Lo que concluimos es que en el modelo es capaz de predecir los movimientos o poses en casi todos los tipos con una precisión del 91% y con un f1-score mayor al 93%, lo que nos dice que el modelo funciona adecuadamente para la tarea.

A continuación se visualiza un poco como se comporta el modelo en una interfaz:









Conclusiones y trabajos futuros

Logros

1. Se desarrolló un sistema funcional capaz de identificar en tiempo real actividades humanas como caminar, girar, sentarse y levantarse con alta precisión.
2. Se construyó una base de datos personalizada a partir de la captura y anotación manual de videos, garantizando control total sobre la calidad y relevancia de los datos.
3. Se implementaron técnicas avanzadas de procesamiento de datos, como imputación de valores nulos, tratamiento de atípicos y reducción de ruido en los landmarks.
4. Se evaluaron diferentes modelos de clasificación (Decision Tree, Random Forest, KNN y SVM), identificando el modelo más robusto para el caso de estudio.
5. Se logró reducir el sobreajuste mediante la optimización de hiperparámetros, alcanzando un F1-score superior al 93% y una precisión general del 91%.
6. Se integró el modelo entrenado en una interfaz visual que permite observar el desempeño del sistema en tiempo real.

Lecciones

1. La calidad del etiquetado de datos es fundamental para el rendimiento del modelo. Un etiquetado riguroso y preciso mejora significativamente la capacidad de generalización del clasificador.
2. El análisis y tratamiento de datos atípicos y nulos es crucial para evitar sesgos y errores durante el entrenamiento.
3. No siempre el modelo con mayor precisión inicial es el más adecuado; la validación cruzada y el ajuste de hiperparámetros ayudan a construir modelos más equilibrados.
4. Herramientas como MediaPipe facilitan la extracción de características relevantes, pero requieren filtrado y procesamiento cuidadoso para ser útiles en clasificación.
5. El uso de técnicas como GridSearchCV permite obtener modelos más estables y confiables para el entorno real.

Trabajos futuros

1. Ampliar la base de datos incorporando más participantes, variabilidad de contextos y nuevos movimientos, para aumentar la robustez y aplicabilidad del modelo.
2. Implementar técnicas de aprendizaje profundo para comparar su desempeño con los modelos tradicionales utilizados.
3. Integrar el sistema con sensores adicionales para enriquecer las entradas del modelo y mejorar la precisión.
4. Desplegar el sistema en dispositivos móviles o entornos clínicos como herramienta de apoyo en rehabilitación física.
5. Añadir capacidades de detección de anomalías o patrones irregulares que puedan alertar sobre caídas u otras situaciones de riesgo.

Referencias

[1] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org

[2] Google, "MediaPipe: Cross-platform, customizable ML solutions for live and streaming media," 2023. [Online]. Available: https://mediapipe.dev/

[3] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[4] J. Hunter, D. Dale, et al., "Matplotlib: A 2D graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. [Online]. Available: https://matplotlib.org