

Human movement analysis: Análisis de actividades humana con detección por video.

Yeison Antonio Rodriguez Zuluaga

Resumen—Este proyecto presenta el desarrollo de un sistema de reconocimiento de actividades humanas basado en landmarks corporales extraídos con MediaPipe y modelos de aprendizaje automático. Se construyó un conjunto de características derivadas de la postura, posición y dinámica del centro de masa. Se implementaron dos modelos finales, uno completo y otro reducido mediante selección de características. Se compararon utilizando métricas de clasificación, análisis por clase, matrices de confusión, curvas ROC, curvas Precision-Recall y medidas de generalización. Finalmente, se discuten limitaciones, posibles mejoras y direcciones para trabajo futuro.

Abstract—This project presents the development of a human activity recognition system based on body landmarks extracted using MediaPipe and machine learning models. A feature set derived from posture, position, and center of mass dynamics was constructed. Two final models were implemented: a full model and a reduced model using feature selection. These models were compared using classification metrics, class analysis, confusion matrices, ROC curves, Precision-Recall curves, and generalization measures. Finally, limitations, potential improvements, and directions for future work are discussed.

I. INTRODUCCIÓN

El reconocimiento automático de actividades humanas es un campo relevante en aplicaciones de salud, deporte, educación y accesibilidad tecnológica. Este proyecto se enfoca en identificar actividades tales como sentarse, pararse, girar, caminar adelante y caminar atrás, utilizando únicamente landmarks del cuerpo extraídos en tiempo real con MediaPipe.

El interés radica en comprender qué tan bien pueden funcionar modelos tradicionales (como Random Forest) cuando la única fuente de información posible es la pose estimada por un modelo 3D, el cual introduce ruido, pérdida de precisión y occlusiones. Además, explorar la reducción de características permite evaluar el compromiso entre rendimiento y complejidad del modelo, lo cual es crítico para sistemas en tiempo real.

II. MARCO TEÓRICO

A. Visión por computador y estimación de poses con MediaPipe Pose

Herramienta de visión por computador que estima 33 puntos corporales en 3D con coordenadas normalizadas. Dichas coordenadas fueron usadas para crear un vector de características usado posteriormente para entrenar modelos de

regresión o clasificación. Este sistema es rápido, sin embargo, puede perder precisión en actividades o movimientos bruscos.

B. Ingeniería de características aplicada

A partir de la estimación de poses usando MediaPipe, Se derivan características importantes tales como: Ángulos articulares, altura corporal, velocidad del centro de masa, posiciones relativas entre puntos clave. Dichas características se relacionan con la postura y el tipo de movimiento y permitan a los modelos analizarlas con mayor facilidad

C. Modelos de clasificación y regresión

Modelos como Random Forest, XGBoost y SVM son adecuados en la clasificación de secuencia de movimientos. En este proyecto dichos modelos se aplican para etiquetar tipos de actividades usando las landmarks extraídas anteriormente.

D. Métricas para clasificación multiclas

Para evaluar el rendimiento se usaron las siguientes métricas: Accuracy, precisión, recall, F1 (macro y weighted), matriz de confusión, AUC-ROC macro y weighted y curvas precisión-recall. Dichas métricas se utilizaron con cada modelo para evaluar su rendimiento y comparar los diferentes modelos entre sí.

III. METODOLOGÍA

A. Recolección de Datos

La recolección de datos empezó con la toma de 50 videos de dos personas, donde cada actividad se grababa 5 veces bajo diferentes condiciones para asegurar variabilidad en los datos. Las 5 actividades llevadas a cabo fueron caminar hacia adelante, caminar hacia atrás, girar, sentarse y pararse. Cada video fue editado para capturar las partes que pertenecían a dicha actividad. Dichos videos se guardaron con la etiqueta “activity_personx_takex”. Donde x correspondían al número de la persona o el numero de la toma, dicho etiquetado en el nombre del archivo se usó para etiquetar los fotogramas en el siguiente paso.

B. Preprocesamiento de Datos

El preprocesamiento de datos inició con la extracción de características usando MediaPipe, primero se trajeron las características de MediaPipe y posteriormente se calcularon características adicionales para realizar un análisis más detallado del movimiento. El análisis de las actividades se hizo fotograma por fotograma, donde cada video se dividió en frames, se trajeron las características relevantes, se calcularon nuevas y se añadieron etiquetas para cada actividad.

C. Integración y Preparación de Datos

Después del cálculo de las landmarks y la creación de dichas landmarks en archivos individuales se combinaron todas en un solo dataset donde cada una de las filas de este correspondía a un fotograma de cada video. Fue este dataset el que se utilizó en los procesos posteriores de entrenamiento. En este paso se usó LabelEncoder para codificar los valores de la columna

actividad de forma que fueran números y permitir el entrenamiento

D. Entrenamiento de modelos de clasificación

Se seleccionaron varios modelos de clasificación supervisada para entrenar el sistema, incluyendo Random Forest, XGBoost y SVM. Estos modelos fueron elegidos por su capacidad para manejar datos estructurados y su eficacia en tareas de clasificación.

Para el entrenamiento de los modelos se generó un script propio de Python que se encargaba de esto. Además de generar las métricas básicas de desempeño para cada modelo.

Para el entrenamiento, se dividió el dataset original en conjunto de entrenamiento y prueba usando la función `train_test_split` de scikit-learn. Esto permite que la evaluación de los modelos se haga con datos diferentes a los vistos durante el entrenamiento. Posteriormente se hace una validación cruzada con los modelos y de esta manera se hace un análisis completo del desempeño de estos bajo diferentes condiciones.

Al final se guardan las métricas de los modelos en un archivo con todos ellos y al evaluar manualmente las métricas se escogió el modelo de RandomForest como el mejor modelo. La métrica para elegir dicho modelo es la precisión. Este modelo se guarda para su uso posterior en la predicción.

E. Reducción de características

Después de haber elegido un modelo, el siguiente paso fue la reducción de características. Se evaluó la importancia de características y se eligieron aquellas que fueran el 90% de importancia para el modelo, después de esto se realizó una comparación entre el modelo original y el modelo reducido, así como se creó un dataset reducido para entrenar el modelo generado.

F. Implementación en tiempo real

Se implementó un sistema de detección de poses en vivo usando CV2, dicho sistema se encargó de capturar en vivo la imagen y de generar los landmarks correspondiente a cada fotograma y realizar predicciones con el modelo que se seleccionó anteriormente.

IV. RESULTADOS

A continuación, se presentan los resultados tanto del entrenamiento de los modelos, como de las métricas del modelo elegido después de hacer la reducción de características.

A. Entrenamiento y Evaluación de Modelos

Se entrenaron y evaluaron tres modelos de clasificación: Random Forest, SVM (Support Vector Machine) y XGBoost, utilizando como criterios de evaluación la precisión. El modelo Random Forest obtuvo una precisión del 99.45% con una desviación estándar de ± 0.00279

, lo que demuestra un desempeño alto y consistente. Por su parte, el modelo SVM alcanzó una precisión del 98,91% con una desviación estándar de ± 0.00292 , mientras que XGBoost logró una precisión del 99.27% con desviación estándar de ± 0.00335

```
model, cv_mean_accuracy, cv_std
RandomForest, 0.9945696800681695, 0.0027968329363165637
SVM, 0.9891375374225916, 0.0029230063678414766
XGBoost, 0.9927588139601646, 0.0033580299146011156
```

Fig. 1. Métricas de los modelos entrenados

B. Reducción de características

Se evaluó el mejor modelo seleccionado, Random Forest y se realizó una selección de características. Al realizar dicha reducción se mantuvieron las características que aportaban en total el 90% del modelo y se entrenó un nuevo modelo con dichas características. Después de entrenar el nuevo modelo se realizó una comparación entre ambos usando la matriz de confusión y la precisión de cada uno.

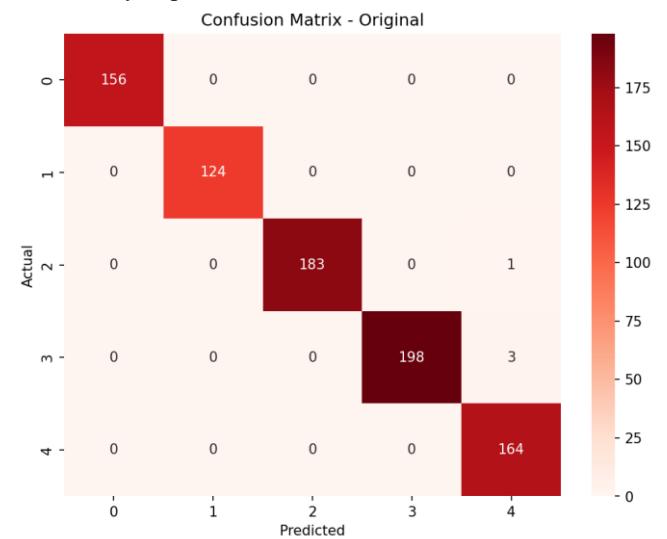


Fig. 2. Matriz de confusión modelo original

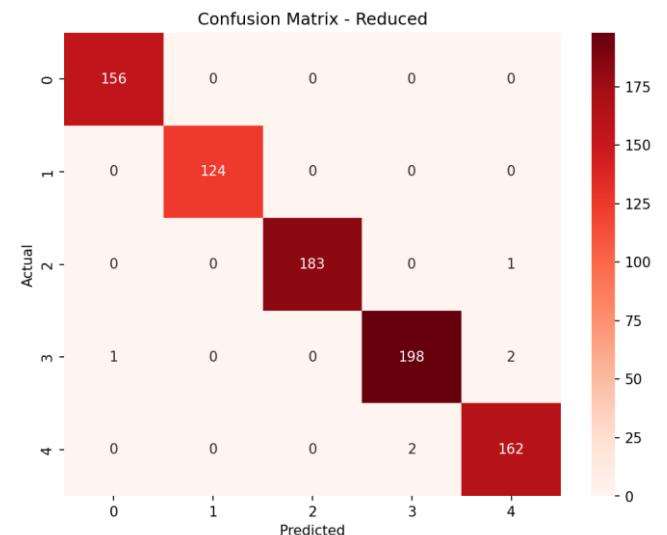


Fig. 3. Matriz de confusión modelo reducido

```
model,accuracy,precision_macro
original,0.9951749095295537,0.9952380952380953
reduced,0.9927623642943305,0.9930897510133179
```

Después de este resultado se llegó a la conclusión de mantener el modelo original, dado que la reducción de características no aportó a la mejora de la precisión, sino que la disminuyó un poco.

C. Resultados de la clasificación en vivo

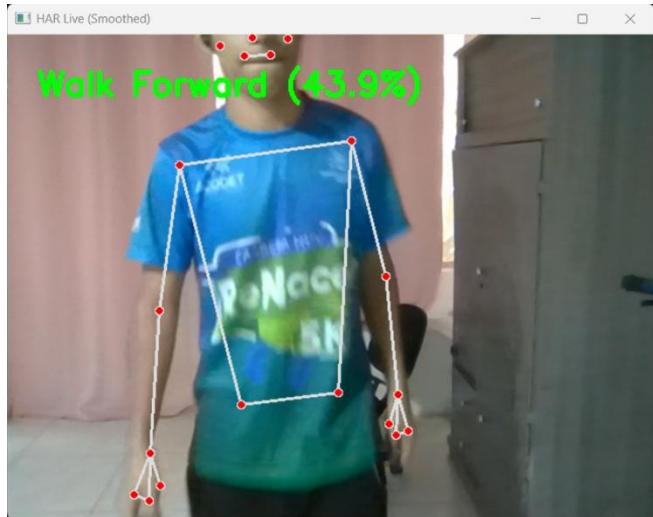


Fig. 2. Detección en Tiempo Real de la Actividad 'Caminar Hacia Adelante'

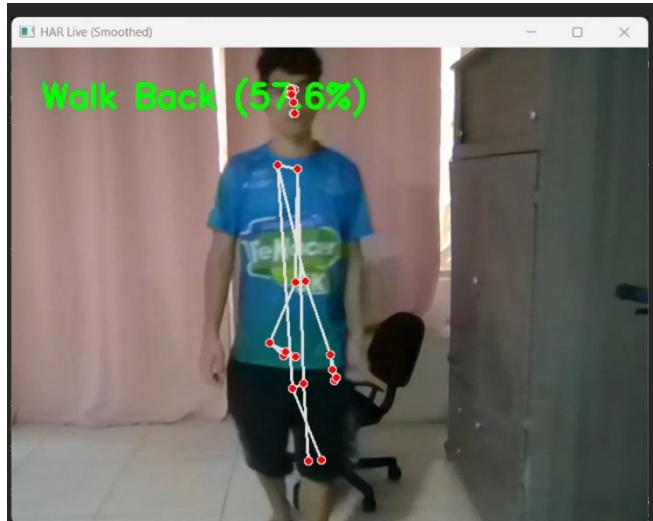


Fig. 3. Detección en Tiempo Real de la Actividad 'Caminar Hacia Atrás'



Fig. 4. Detección en Tiempo Real de la Actividad 'Girar'

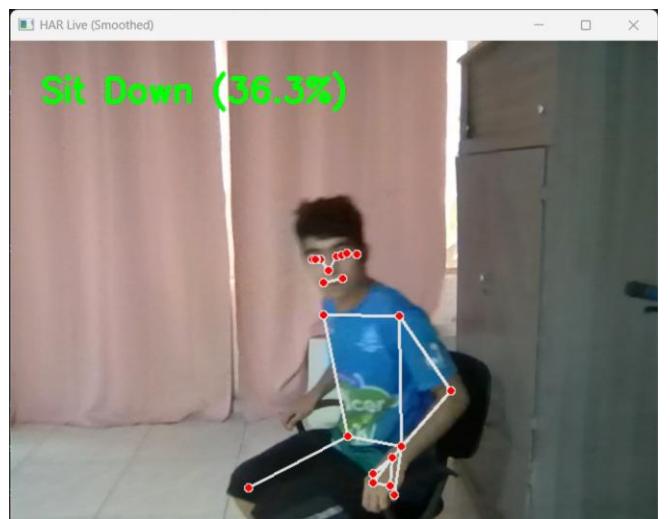


Fig. 5. Detección en Tiempo Real de la Actividad 'Sentarse'

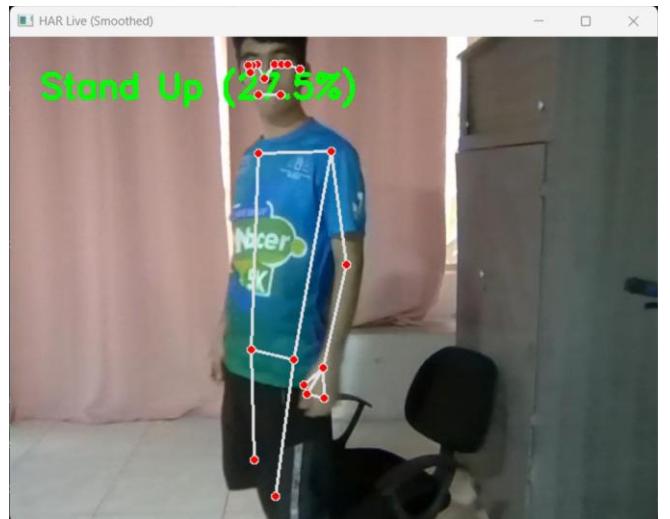


Fig. 6. Detección en Tiempo Real de la Actividad 'Pararse'

El modelo seleccionado, Random Forest, fue implementado en un sistema de detección en tiempo real que utiliza la cámara para capturar y analizar actividades humanas. Para la actividad de caminar hacia adelante (Walk Forward), el sistema alcanzó una confianza del 43%. Para la actividad de caminar hacia atrás

(Walk Back), el sistema alcanzó una confianza del 57,6%. Para la actividad de girar (Turn Around), el sistema alcanzó una confianza del 32,7%. Para la actividad de sentarse (Sit Down), el sistema alcanzó una confianza del 36,3%. Por último, para la actividad de ponerse de pie (Stand Up), el sistema alcanzó una confianza de 27,5%.

V. ANÁLISIS DE RESULTADOS

A continuación, se realiza el análisis de los resultados obtenidos, tanto en el entrenamiento de los modelos como en la detección de actividades en tiempo real.

A. Análisis de los Modelos de Clasificación

Como se mostró en la sección de resultados, los tres modelos alcanzaron unas métricas altas para el problema que se está estudiando, sin embargo, el modelo de Random Forest presentó un desempeño más alto que los otros dos.

El modelo Random Forest superó a los demás porque puede aprender relaciones no lineales entre landmarks, es robusto al ruido natural de Mediapipe, funciona bien con variables heterogéneas, tolera correlación interna, maneja clases difíciles y generaliza mejor gracias al ensamble de árboles. Además, proporciona métricas interpretables de importancia de características, facilitando la etapa de reducción de dimensionalidad.

Por lo anterior, el modelo seleccionado para el análisis en tiempo real fue el modelo de Random Forest.

B. Análisis de Resultados en Tiempo Real

El modelo Random Forest fue implementado en el módulo de captura en vivo, donde se evaluó su desempeño al clasificar actividades humanas utilizando la cámara del sistema. Los resultados de la confianza de cada actividad fueron:

- Walk Forward: 43%
- Walk Back: 57.6%
- Turn Around: 32.7%
- Sit Down: 36.3%
- Stand Up: 27.5%

Estos valores muestran que, en un entorno real y dinámico, la confianza del modelo es significativamente menor que la reportada durante el entrenamiento y validación. Esta diferencia podría deberse a varios factores:

- Variabilidad en el entorno real: Las condiciones de iluminación, fondo, la distancia a la cámara, la diferencia de la cámara usada para capturar los videos de entrenamiento afecta la detección de landmarks y podrían introducir ruido adicional.
- Datos de entrenamiento controlados: Los datos de entrenamiento fueron controlados específicamente para que se vieran de la mejor manera. Durante la captura en vivo

podrían aparecer variaciones que los modelos no habían visto en los datos de entrenamiento.

- Actividades con frames similares: Dado que la manera para clasificar las actividades fueron frames, es posible que, al capturar en vivo, se hallen posturas intermedias que son difíciles de distinguir entre sí.

Los resultados muestran que el modelo generaliza parcialmente en el entorno real, pero con una confianza menor a la observada en el entrenamiento, además que confunde actividades con mayor frecuencia a la que se había visto.

VI. CONCLUSIONES

El proyecto logró implementar un sistema completo de reconocimiento de actividades humanas mediante visión por computadora, desde la recolección y preparación de datos hasta el despliegue de un modelo funcional en tiempo real. A través del análisis comparativo de varios modelos de clasificación, Random Forest demostró ser la opción más robusta para este problema.

El análisis en tiempo real evidenció que, aunque el modelo mantiene un nivel de funcionamiento mínimo, su confianza disminuye de manera drástica frente a condiciones nuevas o no controladas anteriormente. Esto demuestra una brecha entre el rendimiento en el conjunto de prueba y el rendimiento en escenarios reales, así como también una posible recolección pobre de datos que impidió al modelo generalizar adecuadamente bajo diferentes condiciones.

A pesar de estas limitaciones, el sistema logró identificar las actividades en algunos casos, mostrando su potencial como herramienta, pero también su necesidad de hacer un pulido y un análisis adecuado desde las primeras etapas del proyecto.

A pesar de lo anterior este proyecto sirvió como un primer acercamiento a la integración de herramientas de inteligencia artificial para resolver un problema real, así como la importancia de un proceso adecuado desde las primeras etapas del desarrollo.

El sistema desarrollado cumple de forma mínima con los requerimientos solicitados, por lo tanto, hay muchos aspectos que pueden mejorarse. En primer lugar, como se mencionó anteriormente una selección más cuidadosa y variada de datos podría permitir a los modelos funcionar en condiciones de captura en vivo bajo situaciones inesperadas. Segundo, se podría incorporar un mayor número de actividades que haga el sistema más versátil. Finalmente, la mejora de la interfaz de usuario, hacerla más intuitiva o adaptar el sistema para su uso en diferentes dispositivos y bajo diferentes circunstancias.

REFERENCIAS

- [1] Google AI, «Guía de soluciones de MediaPipe,» 26 Febrero 2025. [En línea]. Available: <https://ai.google.dev/edge/mediapipe/solutions/guide?hl=es-419>. [Último acceso: 8 Junio 2025].
- [2] T. C. a. C. Guestrin, «XGBoost: A Scalable Tree Boosting System,» 13 Agosto 2016. [En línea]. Available: <https://dl.acm.org/doi/10.1145/2939672.2939785>. [Último acceso: 08 Junio 2025].
- [3] OpenCV Developers, «OpenCV Documentation index,» 02 Junio 2025. [En línea]. Available: [view-source:https://docs.opencv.org](https://docs.opencv.org). [Último acceso: 08 Junio 2025].