Sistema de Análisis de Actividades Humanas Basado en Inteligencia Artificial

Santiago Barraza
Sebastian Lopez Garcia
Juan Sebastian Medina

Resumen: ***Este proyecto desarrolla un sistema de análisis de actividades humanas utilizando inteligencia artificial. Se analizan movimientos específicos como sentarse, ponerse de pie y caminar mediante la extracción de características biomecánicas de videos grabados, incluyendo velocidad articular, ángulos articulares y deltas de movimiento. Utilizando herramientas como MediaPipe y CVAT, los datos se anotaron y procesaron para entrenar un modelo de clasificación supervisada. El sistema está diseñado para ofrecer información precisa en tiempo real sobre posturas y movimientos, con aplicaciones en rehabilitación, análisis deportivo y monitoreo de actividades cotidianas.***

Abstract: ***This project develops a human activity analysis system using artificial intelligence. Specific movements such as sitting, standing, and walking are analyzed by extracting biomechanical features from recorded videos, including joint velocity, joint angles, and motion deltas. Using tools such as MediaPipe and CVAT, the data was annotated and processed to train a supervised classification model. The system is designed to provide accurate real-time information on postures and movements, with applications in rehabilitation, sports analysis, and monitoring of daily activities.***

I. INTRODUCCIÓN

La capacidad de clasificar actividades humanas tiene aplicaciones importantes en sectores como la salud, el deporte y la seguridad. Este proyecto busca implementar un sistema basado en inteligencia artificial que pueda reconocer actividades específicas a partir de datos visuales y proporcionar información biomecánica relevante. Clasificar actividades humanas presenta desafíos relacionados con la precisión, la variabilidad en las posturas y el ruido en los datos. La implementación de sistemas que analicen las actividades humanas no solo tiene el potencial de mejorar la rehabilitación física y optimizar el rendimiento deportivo, sino también de asistir en el monitoreo de personas mayores o pacientes, con aplicaciones significativas en distintas áreas del conocimiento, especialmente en la medicina.

II. MARCO TEÓRICO

A. Biomecánica Articular:

Las velocidades y ángulos articulares son métricas fundamentales para diferenciar actividades dinámicas de estáticas. Estas características permiten un análisis más detallado del movimiento y la postura.

B. Normalización:

Se intentó realizar el ajuste de los datos con respecto a las caderas eliminando dependencias relacionadas con el ángulo de grabación y la altura del sujeto. Este proceso asegura que las características extraídas sean más consistentes y comparables entre individuos.

Posteriormente utilizando un estándar scaler para asegurar la normalización de todos los datos en el dataset.

C. Modelos Supervisados:

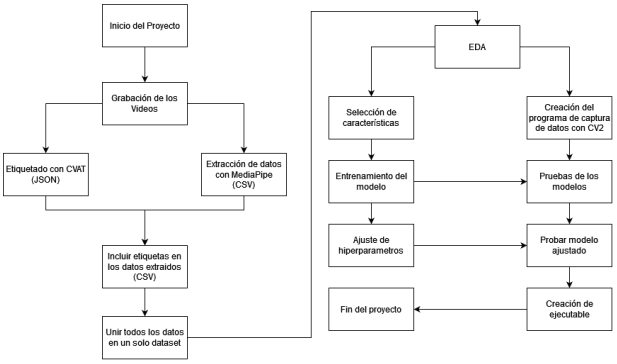
Técnicas como XGBoost son especialmente útiles para problemas de clasificación debido a su capacidad para manejar datos estructurados y mitigar el impacto del ruido. Este enfoque garantiza un rendimiento robusto en tareas de predicción.

D. Hiperparámetros:

La optimización de los hiperparámetros es clave para mejorar la precisión del modelo y evitar problemas como el sobreajuste o el subajuste. Este proceso implica ajustar valores clave que controlan el aprendizaje del modelo, como la profundidad de los árboles o la tasa de aprendizaje.

III. METODOLOGÍA

A continuación se presenta el flujo de trabajo seguido durante el proyecto:

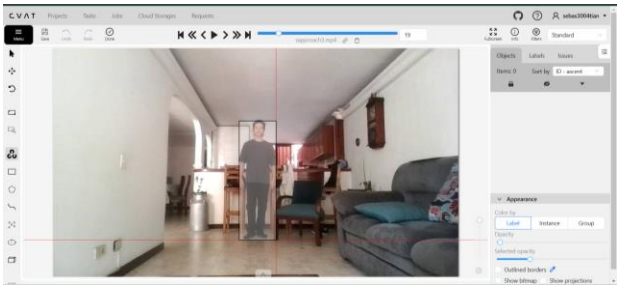


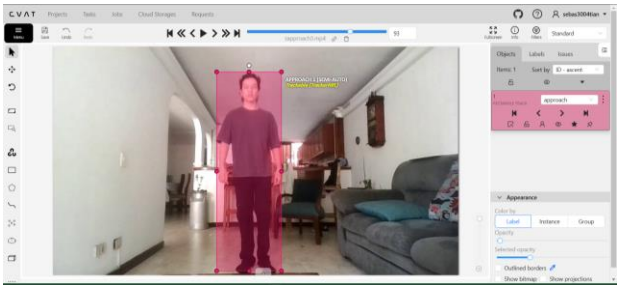
A. Recolección de Datos:

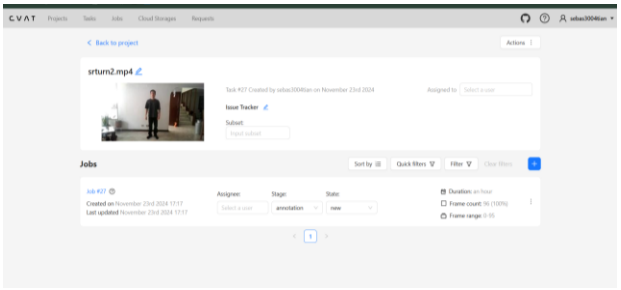
La recolección de datos fue un proceso clave para garantizar un conjunto de datos robusto y bien estructurado, fundamental para el entrenamiento y evaluación del modelo. Este proceso incluyó:

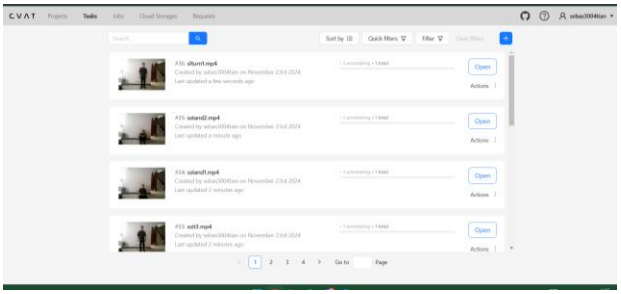
- **Grabación de Actividades**
  Se registraron un total de 78 videos que capturaron 9 gestos específicos desde múltiples perspectivas. Los videos se grabaron con una diversidad de fondos, personas y vestuarios, con el objetivo de generar un conjunto de datos representativo y diverso.
- **Segmentación y Extracción de Características**
  Cada video fue segmentado en 7,588 frames, y para cada uno se extrajeron 99 características que describen las posiciones de las articulaciones. Este nivel de detalle permitió estructurar los datos de manera que reflejaran con precisión los movimientos y las posturas involucradas.
- **Anotación con CVAT**
  Se utilizó la herramienta CVAT (Computer Vision Annotation Tool) para analizar los datos y asignar etiquetas precisas a los frames correspondientes. Este paso fue crucial para garantizar que cada gesto estuviera correctamente identificado, facilitando la calidad del entrenamiento del modelo.

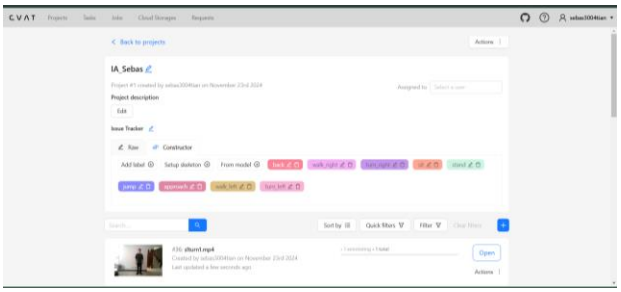
A continuación, se presentan imágenes del proceso de etiquetado utilizando CVAT, destacando la asignación de etiquetas a los gestos en los videos grabados.











B. Procesamiento de Datos:

- Extracción de landmarks con MediaPipe para caderas, rodillas y tobillos:

```
frame,landmark,x,y,z,visibility
0,NOSE,0.526887834072113,0.34001630544662476,-0.5765976309776306,0.9999985694885254
0,LEFT_EYE_INNER,0.5351219177246094,0.33378681540489197,-0.5559157729148865,0.9999969005584717
0,LEFT_EYE,0.5402572154998779,0.33373337984085083,-0.5563306808847168,0.9999960660934448
0,LEFT_EYE_OUTER,0.545035183429718,0.33385491371154785,-0.5562533736228943,0.9999971389770508
0,RIGHT_EYE_INNER,0.5216906666755676,0.3339767754079114,-0.5579906105995178,0.9999978542327881
0,RIGHT_EYE,0.5174670219421387,0.3340131640434265,-0.5583615899085999,0.9999980926513672
0,RIGHT_EYE_OUTER,0.5141100883483887,0.3340322971343994,-0.5585408806800842,0.9999985694885254
0,LEFT_EAR,0.5496150851249695,0.3370426893234253,-0.39686718583106995,0.9999957084655762
0,RIGHT_EAR,0.5116192698478699,0.3355888426303863 5,-0.4061176180839 5386,0.9999911785125732
0,MOUTH_LEFT,0.5359665751457214,0.34695008397102356,-0.5113421678543091,0.9999978542327881
```

- Normalización respecto a las caderas para eliminar variaciones externas:

  Se usan los CSV con las coordenadas corporales,

normalizando los datos en relación al centro de las caderas para estandarizar las posiciones. Cada archivo se ajusta multiplicando las coordenadas por un factor uniforme y alineándose respecto al centro de las caderas. Los archivos resultantes se guardan organizadamente en una carpeta de salida, listos para ser procesados.

- Integración de datos de landmarks con datos etiquetados con CVAT para cada video en un solo dataset dividiendo los registros por frames, agrupando todos los registros de movimientos, en un solo CSV.

```
movement_data = pd.read_csv("output_with_angles.csv")
movement_data
```
✓ 0.1s

FT_FOOT_INDEX_x	LEFT_HEEL_x	LEFT_HIP_x	...	RIGHT_PINKY_x	RIGHT_SHOULDER_z	RIGHT_THUMB_z	RIGHT_WRIST_z	annotation	videoid
0.570951	0.513027	0.484337	...	0.236498	0.325306	0.253600	0.266685	Still	bapproach1
0.567500	0.512050	0.484133	...	0.247105	0.331144	0.262427	0.275492	Still	bapproach1
0.565493	0.510918	0.481966	...	0.245681	0.328713	0.261951	0.274716	Still	bapproach1
0.564397	0.510423	0.479499	...	0.231729	0.321172	0.248262	0.261434	Still	bapproach1
0.564309	0.509886	0.475675	...	0.229942	0.321479	0.246645	0.259679	Still	bapproach1
...	...	...	...	...	...	...	...	...	...
0.764315	0.626971	0.614757	...	0.338444	0.450269	0.361474	0.374412	stand	sstand2
0.764197	0.625224	0.607706	...	0.373864	0.482316	0.396372	0.409577	stand	sstand2
0.764177	0.622663	0.600839	...	0.388165	0.502106	0.410555	0.423952	stand	sstand2
0.764033	0.622570	0.595394	...	0.398608	0.511323	0.421087	0.434140	stand	sstand2
0.764365	0.622512	0.588518	...	0.403635	0.523499	0.427123	0.439893	stand	sstand2

- Cálculo de métricas:
    - Velocidades articulares.
    - Ángulos en las rodillas.
    - Distancia entre los hombros
    - Deltas de movimiento entre frames consecutivos.

  El cálculo de métricas como velocidades articulares, ángulos en las rodillas, distancias entre los hombros y deltas de movimiento entre frames consecutivos se realizó para enriquecer los datos y extraer información clave sobre la dinámica y la postura del movimiento humano. Las velocidades articulares reflejan la rapidez y fluidez de las articulaciones, lo que es fundamental para evaluar patrones dinámicos en actividades específicas. Los ángulos en las rodillas ofrecen una medida precisa de la flexión o extensión, permitiendo identificar posturas y posibles anomalías. La distancia entre los hombros proporciona información sobre la estabilidad y simetría del torso, lo que es relevante para analizar el equilibrio corporal. Finalmente, los deltas de movimiento entre frames capturan los cambios entre momentos consecutivos, permitiendo modelar las transiciones y detectar patrones temporales. Estos cálculos son esenciales para generar un conjunto de datos detallado y representativo que permita al modelo aprender de forma más efectiva y realizar predicciones precisas basadas en movimientos y posturas.

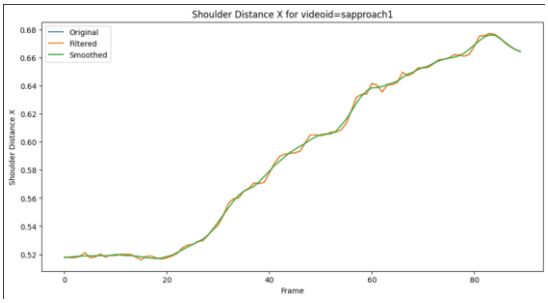
- Filtrado:

  Se aplicó filtrado de datos para eliminar ruido y mejorar la calidad de las características extraídas de los datos de movimiento. En este caso, se utilizaron dos métodos de filtrado complementarios: el filtro **Butterworth** y el filtro **Savitzky-Golay**.

  - **Filtro Butterworth:** Este filtro es un tipo de filtro pasa-bajas que suaviza las señales al atenuar las frecuencias altas, las cuales suelen estar asociadas con ruido. Se utiliza un filtro de orden 3 con una frecuencia de corte normalizada de 0.933333, lo que permite conservar la mayoría de las características dinámicas del movimiento humano mientras se elimina el ruido indeseado. Este filtro es ideal porque tiene una respuesta en frecuencia suave, lo que evita artefactos como ondulaciones en la señal procesada. Su implementación, mediante la función filtfilt, aplica el filtrado hacia adelante y hacia atrás, asegurando que no haya desfase en las señales procesadas.
  - **Filtro Savitzky-Golay:** Este filtro se aplica después del filtrado Butterworth para una suavización adicional. A diferencia de los filtros convencionales, Savitzky-Golay ajusta polinomios locales a las ventanas deslizantes de la señal, lo que permite preservar mejor las características originales, como picos y valles, mientras suaviza las oscilaciones menores. En este caso, se utiliza una ventana de longitud 11 y un polinomio de orden 3, parámetros que equilibran la suavización y la fidelidad a los datos originales.

```
from scipy.signal import butter, filtfilt, savgol_filter

# Crear los coeficientes del filtro Butterworth
b, a = butter(N=3, Wn=0.933333, btype='low')

# Inicializar DataFrames finales para los datos filtrados y suavizados
filtered_dataset = movement_data.copy()
smoothed_dataset = movement_data.copy()

# Obtener los IDs únicos de los videos
video_ids = movement_data["videoid"].unique()

# Procesar cada video
for video_id in video_ids:
    # Filtrar los datos por el ID del video actual
    video_data = movement_data[movement_data["videoid"] == video_id]
    # Identificar columnas numéricas (exceptuando 'frame')
    numeric_columns = video_data.select_dtypes(include=['number']).columns
    columns_to_filter = [col for col in numeric_columns if col != 'frame']
    # Crear copia de los datos originales para aplicar los filtros
    filtered_data = video_data.copy()
    # Aplicar el filtro Butterworth a las columnas seleccionadas
    for column in columns_to_filter:
        filtered_data[column] = filtfilt(b, a, video_data[column])
    # Crear copia para aplicar el filtro Savitzky-Golay
    smoothed_data = filtered_data.copy()
    for column in columns_to_filter:
        smoothed_data[column] = savgol_filter(filtered_data[column], window_length=11, polyorder=3)
    # Reemplazar los datos filtrados y suavizados en los DataFrames finales
    filtered_dataset.loc[video_data.index, columns_to_filter] = filtered_data[columns_to_filter]
    smoothed_dataset.loc[video_data.index, columns_to_filter] = smoothed_data[columns_to_filter]
```

A continuación, se presenta una comparación de los métodos de filtrado aplicados a un video específico (sapproach1), en el que se observa un acercamiento (approach). En la gráfica resultante, se superponen tres trazos que corresponden a diferentes conjuntos de datos: los datos originales, los datos filtrados con un filtro pasa-bajas, y los datos suavizados, obtenidos mediante la aplicación conjunta de los filtros Butterworth y Savitzky-Golay. Para cada conjunto, se calcula la distancia entre los hombros en el eje X, una métrica clave para analizar el movimiento, la postura y los cambios en la profundidad de la cámara.



Esta visualización ilustra cómo el proceso de filtrado mejora la claridad y estabilidad de los datos, eliminando irregularidades menores y preservando las características más relevantes del movimiento. Al comparar las tres señales, se observa cómo el filtrado contribuye a hacer que las señales sean más estables y representativas del movimiento real, lo que resulta fundamental para un análisis más preciso y para el entrenamiento del modelo.

C. Análisis Exploratorio:
- Visualización de distribuciones mediante boxplots.
- Identificación de patrones distintivos para cada actividad basada en métricas calculadas.

D. Entrenamiento del Modelo:
- Validación cruzada para evitar sobreajuste.
- Ajuste de hiperparámetros con GridSearch.
- Evaluación con precisión, recall y F1-score.

Se realizaron modelos con distintos conjuntos de atributos, en primer lugar se realizó un entrenamiento con datos de entrenamiento de posiciones de distintas marcas corporales, y posteriormente se redefinió el enfoque a uno con cambios en la distancia de las marcas, es decir usando deltas entre las distintas posiciones registradas.

Se realizaron pruebas para los datos normalizados; dejando resultados de un poco más de 0.97 para cada métrica.

```
# Normalized
x = movement_data.drop(['annotation', 'videoid', 'frame'], axis=1)

X_train, X_test, y_train, y_test = train_test_split(x, encoded_labels, test_size=0.2, random_state=42)

model = XGBClassifier()

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = model.score(X_test, y_test)
# F1-score
f1 = f1_score(y_test, y_pred, average='weighted')  # Cambiar a 'micro', 'macro' o 'binary' según el caso

# Recall
recall = recall_score(y_test, y_pred, average='weighted')  # Cambiar a 'micro', 'macro' o 'binary' según el caso

# Print results
print(f"Accuracy: {accuracy}")
print(f"F1-Score: {f1}")
print(f"Recall: {recall}")
✓ 2.0s
Accuracy: 0.9756258234519104
F1-Score: 0.9754799518652l7
Recall: 0.9756258234519104
```

Luego al conjunto de datos se le realizó la misma prueba pero aplicando al conjunto un filtro pasabajas para restringir las frecuencias superiores a 15, y aunque el modelo mejoró sus puntuaciones, no fue efectivo a la hora de hacer pruebas en tiempo real.

```
# Normalized + Filtered
x = filtered_dataset.drop(['annotation', 'videoid', 'frame'], axis=1)

X_train, X_test, y_train, y_test = train_test_split(x, encoded_labels, test_size=0.2, random_state=42)

model = XGBClassifier()

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

accuracy = model.score(X_test, y_test)
f1 = f1_score(y_test, y_pred, average='weighted')

# Recall
recall = recall_score(y_test, y_pred, average='weighted')

# Print results
print(f"Accuracy: {accuracy}")
print(f"F1-Score: {f1}")
print(f"Recall: {recall}")
✓ 24s
Accuracy: 0.9808959156785244
F1-Score: 0.9808546572140904
Recall: 0.9808959156785244
```

IV. RESULTADOS

A. Modelo entrenado con posiciones de articulaciones:

Al entrenar el modelo con posiciones en x, y, z de las articulaciones rápidamente nos dimos cuenta que tendía a generar overfitting pues los movimiento que queríamos analizar no deberían depender si los hacemos desde algún punto específico de la imagen.

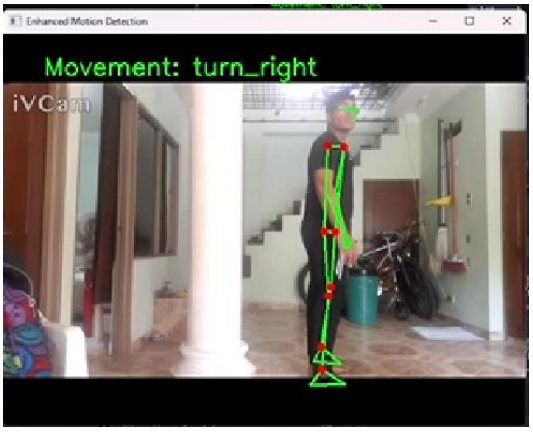
Al entrenar este modelo y hacer gridsearch obtuvimos una accuracy de 98% con datos de prueba, sin embargo al probar nuestro modelo en tiempo real no proveía información relevante frente al movimiento realizado.

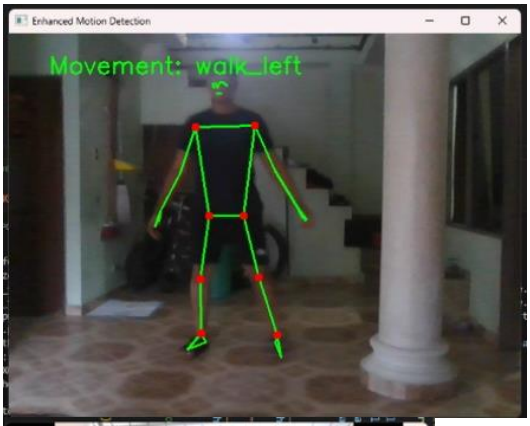
B. Modelo entrenado con valores calculados a partir de posiciones.

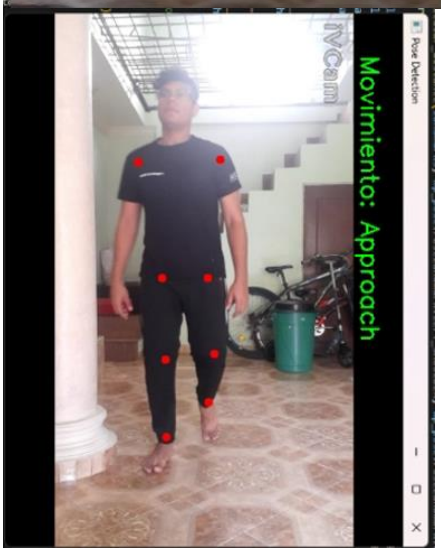
El uso de las features derivadas, como ángulos entre articulaciones, distancias relativas y parámetros dinámicos como velocidades y aceleraciones, ofreció resultados significativamente más robustos en comparación con los landmarks de posición absoluta (x, y, z). Estas features encapsulan información esencial del movimiento, eliminando la dependencia del sistema de coordenadas de la imagen y reduciendo el riesgo de overfitting. Por ejemplo, valores como LEFT_KNEE_ANGLE o hip_distance son intrínsecos al movimiento y no varían según el punto de partida de la acción, permitiendo que el modelo se enfoque en patrones del movimiento más universales. Además, métricas dinámicas como center_velocity_x o step_length_rolling_std reflejan cambios continuos en las acciones, haciendo que el modelo sea más generalizable. Con estas features, el modelo no solo mejoró su desempeño en métricas tradicionales como accuracy (alcanzando 91.96% tras GridSearch), sino que también mostró un comportamiento más coherente en pruebas en tiempo real, proporcionando información relevante para las acciones realizadas.

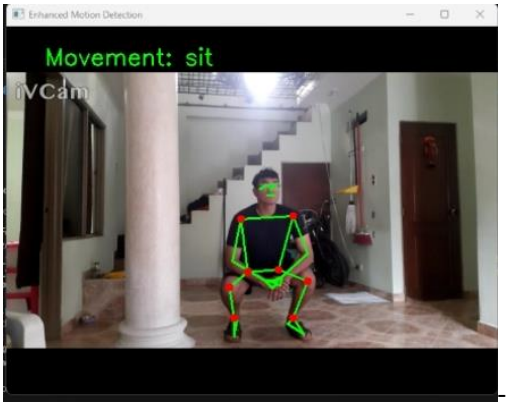
```
Best Parameters: {'colsample_bytree': 0.6, 'learning_rate': 0.2, 'max_depth': 8, 'n_estimators': 300, 'subsample': 1.0}
Best Score: 0.90
Test Set Accuracy: 91.96%

Classification Report:
              precision    recall  f1-score   support

       Still       0.87      0.92      0.90       558
    approach       0.97      0.95      0.96       213
        back       0.98      0.97      0.98       272
        jump       0.91      0.86      0.88        57
   turn_left       0.93      1.00      0.97        43
   walk_left       1.00      0.97      0.98        65
  turn_right       0.96      0.93      0.95        56
  walk_right       0.98      0.99      0.98        93
         sit       0.92      0.90      0.91       112
       stand       0.53      0.35      0.42        49

    accuracy                           0.92      1518
   macro avg       0.91      0.88      0.89      1518
weighted avg       0.92      0.92      0.92      1518
```

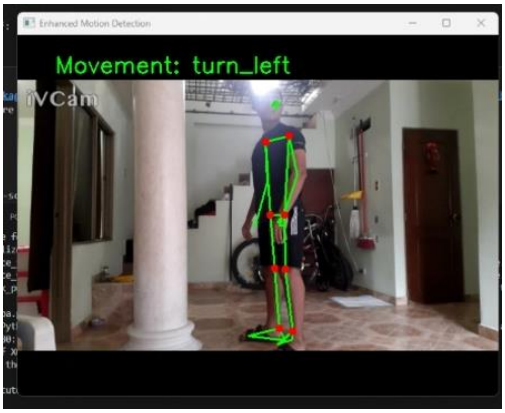
C. Visualización de resultados:











V. CONCLUSIONES Y TRABAJO FUTURO

A. Logros:

En este proyecto, se desarrolló un sistema eficiente y funcional para la clasificación de actividades humanas utilizando inteligencia artificial, específicamente un modelo de aprendizaje automático entrenado con datos biomecánicos. El sistema logró identificar y diferenciar con precisión una variedad de actividades y movimientos corporales, incluyendo girar hacia la izquierda (turn left), girar hacia la derecha (turn right), caminar hacia la izquierda (walk left), caminar hacia la derecha (walk right), saltar (jump), acercarse (approach), retroceder (back), sentarse (sit), y ponerse de pie (stand). Esto representa un avance significativo en las bases para aplicaciones en áreas como la rehabilitación física, el monitoreo de actividad física y la interacción humano-computadora.

B. Lecciones aprendidas:

Durante el desarrollo del proyecto, se identificaron varios aspectos clave que influenciaron el desempeño del sistema y ofrecen valiosas enseñanzas para trabajos futuros:

- **Dificultad en la identificación de movimientos específicos:** Uno de los mayores retos fue garantizar que el modelo

diferenciara correctamente entre movimientos similares, como girar hacia la izquierda o caminar hacia la izquierda. Estos problemas iniciales destacaron la necesidad de enriquecer las características utilizadas en el modelo. Fue crucial implementar métricas adicionales derivadas de las posiciones y ángulos de las articulaciones, como las distancias relativas entre puntos clave (ej., hombros y caderas) y los ángulos en rodillas y caderas. Estas métricas permitieron capturar con mayor precisión las dinámicas específicas de cada movimiento.

- **Importancia de las características biomecánicas calculadas:** El simple uso de coordenadas no fue suficiente para representar con claridad la complejidad de los movimientos humanos. La inclusión de cálculos biomecánicos avanzados, como los ángulos articulares y las distancias relativas entre puntos clave, resultó fundamental para mejorar la capacidad del modelo de distinguir entre diferentes actividades. Esto subraya la importancia de diseñar cuidadosamente las características en proyectos similares, asegurando que reflejen con precisión la dinámica corporal.

- **Necesidad de datos diversos:** A pesar de las mejoras en las características, se evidenció que la limitada diversidad del conjunto de datos restringió la capacidad del modelo para generalizar. Los datos utilizados se grabaron bajo condiciones controladas y con un número reducido de participantes, lo que redujo la robustez del sistema frente a variaciones en escenarios reales. Esto reafirmó la necesidad de incorporar datos provenientes de grabaciones en entornos variados, con diferentes condiciones de iluminación, perspectivas de cámara y poblaciones diversas, para garantizar un desempeño más robusto y generalizable.

C. Mejoras futuras:

Para construir sobre los logros alcanzados y abordar las limitaciones identificadas, se propone un plan de mejora que incluye las siguientes acciones:

- **Ampliación del conjunto de datos**: Es fundamental expandir significativamente el dataset utilizado para el entrenamiento y la validación del modelo. Esto implicaría la grabación de nuevos videos bajo diversas condiciones, tales como distintos ángulos de cámara, escenarios más variados, cambios en la iluminación y grabaciones en interiores y exteriores. Además, incluir personas de diferentes edades, alturas, complexiones físicas y niveles de habilidad motriz podría mejorar la capacidad del modelo para generalizar a una población más amplia.

- **Exploración de arquitecturas avanzadas de redes neuronales profundas**: Aunque los modelos actuales basados en aprendizaje automático han demostrado ser efectivos, explorar redes neuronales profundas, como las redes convolucionales (CNNs) o las redes neuronales recurrentes (RNNs), podría mejorar significativamente el desempeño en la clasificación de actividades más complejas o con movimientos más sutiles. Estas arquitecturas tienen el potencial de extraer automáticamente características relevantes y capturar mejor las relaciones temporales en los movimientos.

- **Validación en escenarios reales**: Evaluar el sistema en aplicaciones prácticas, como en rehabilitación física, deportes o vigilancia, podría proporcionar información valiosa sobre su desempeño en condiciones no controladas. Esto permitiría realizar ajustes adicionales para optimizar su utilidad en contextos reales.

- **Integración de sensores adicionales**: Incorporar datos provenientes de sensores inerciales (IMUs) o dispositivos portátiles podría complementar la información visual y mejorar la precisión en escenarios donde las cámaras enfrentan limitaciones, como oclusiones o poca luz.

VI. REFERENCIAS

[1] N. Jirafe, "How to filter noise with a low pass filter — Python", Analytics Vidhya, Dec. 27, 2019. [Online]. Available: https://medium.com/analytics-vidhya/how-to-filter-noise-with-a-low-pass-filter-python-885223e5e9b7

[2]"MediaPipe Vs OpenPose - QuickPose.ai," *QuickPose.ai*, Jul. 26, 2024. https://quickpose.ai/faqs/mediapipe-vs-openpose/ (accessed Nov. 24, 2024).