Informe Final: Desarrollo de un Sistema de Reconocimiento de Actividades Humanas y Seguimiento Articular en Tiempo Real

Facultad de Ingeniería, Diseño y Ciencias Aplicadas Departamento de Computación y Sistemas Inteligentes Ingeniería de Sistemas

---

Docentes: Uram Sosa Aguirre, Milton Sarria Paja
Asignatura: Inteligencia Artificial I
Institución: Universidad Icesi

Integrantes:

Daniel Esteban Jaraba Gaviria, Santiago Angel Ordoñez, Santiago Gutiérrez Villegas

Día de Entrega:

13/Junio/2025

Índice:

Resumen

Este proyecto aborda el desarrollo de un sistema de software para el reconocimiento de actividades humanas (HAR) y el análisis postural en tiempo real. Utilizando la metodología CRISP-DM, se diseñó e implementó una solución completa que captura video a través de una cámara, procesa los fotogramas para extraer un esqueleto de 18 articulaciones clave mediante la librería MediaPipe, y clasifica la actividad ejecutada por una persona en una de cinco categorías predefinidas: caminar hacia la cámara, caminar de regreso, girar, sentarse y levantarse. El núcleo del sistema es un modelo de aprendizaje supervisado, específicamente un Support Vector Machine (SVM), entrenado con características biomecánicas como ángulos articulares, velocidades y la inclinación del tronco. El sistema final ofrece una visualización en vivo que superpone el esqueleto detectado, la actividad clasificada con su nivel de confianza y los cuadros por segundo (FPS) del procesamiento, demostrando la viabilidad de aplicar técnicas de visión por computador y machine learning para el análisis de movimiento con baja latencia.

Introducción

El Reconocimiento de Actividades Humanas (HAR) es un campo de investigación prominente dentro de la inteligencia artificial y la visión por computador, con aplicaciones que van desde la rehabilitación médica y el análisis deportivo hasta la interacción humano-robot y los sistemas de vigilancia inteligente. La capacidad de un sistema para entender y cuantificar el movimiento humano en tiempo real abre un abanico de posibilidades para crear herramientas de apoyo más intuitivas y contextuales.

Descripción del Problema: El objetivo de este proyecto es desarrollar una herramienta de software capaz de analizar y clasificar cinco actividades humanas básicas (caminar hacia la cámara, caminar de regreso, girar, sentarse y ponerse de pie) a partir de una fuente de video en tiempo real. Además de la clasificación, el sistema debe realizar un seguimiento de movimientos articulares y parámetros posturales clave, como la inclinación del tronco y los ángulos de las rodillas. El principal reto técnico reside en procesar la secuencia de poses, extraídas cuadro a cuadro, para realizar una clasificación supervisada y un análisis de serie temporal bajo estrictas restricciones de baja latencia que permitan una respuesta fluida e interactiva.

Relevancia y Contexto: La importancia de un sistema como este radica en su potencial para ofrecer retroalimentación biomecánica inmediata y accesible. En contextos de fisioterapia, podría guiar a los pacientes en la correcta ejecución de sus ejercicios; en ergonomía, podría alertar sobre posturas perjudiciales en un entorno de oficina; y en deportes, podría ayudar a los atletas a optimizar su técnica.

Consideraciones Éticas: Desde su concepción, el proyecto se ha guiado por principios éticos para garantizar un manejo responsable de los datos. Las medidas implementadas incluyen la obtención de consentimiento informado de todos los participantes grabados, la protección de la privacidad mediante la no captura de rostros o información identificable, el uso restringido del material de video exclusivamente para fines académicos del curso, la promoción de la diversidad para evitar sesgos en el modelo y el almacenamiento seguro del dataset.

Fundamentos Teóricos

Para comprender el desarrollo del proyecto, es necesario conocer los siguientes conceptos clave:

- **Estimación de Pose Humana (Human Pose Estimation):** Es una técnica de visión por computador que detecta y localiza las articulaciones clave (landmarks) del cuerpo de una

persona en imágenes o videos. Para este proyecto, se utilizó MediaPipe Pose, una solución de Google que ofrece un seguimiento robusto y de baja latencia de 33 landmarks corporales en 3D, ideal para aplicaciones en tiempo real. Nuestro sistema se enfoca en un subconjunto de 18 de estas articulaciones para optimizar el procesamiento.

- **Ingeniería de Características (Feature Engineering):** Las coordenadas crudas (x, y, z) de las articulaciones, aunque informativas, no son suficientes para que un modelo distinga actividades complejas. Por ello, se realiza la ingeniería de características para crear atributos más discriminativos. En este proyecto, se calcularon:
    - Ángulos articulares: Como el ángulo de la rodilla y la inclinación del torso, para capturar la postura del cuerpo.
    - Velocidades y aceleraciones: De las muñecas y tobillos para describir la dinámica del movimiento.
- **Metodología CRISP-DM:** Es un estándar de la industria para gestionar proyectos de minería de datos y analítica. El proyecto se adhirió a sus seis fases (comprensión del negocio, comprensión de los datos, preparación de datos, modelado, evaluación y despliegue) para asegurar un desarrollo estructurado, iterativo y orientado a objetivos.
- **Modelos de Clasificación Supervisada:** Se emplean algoritmos que aprenden a partir de datos previamente etiquetados. Aunque se consideraron varios modelos como Random Forest y XGBoost, la implementación final utiliza un Support Vector Machine (SVM), un clasificador potente y eficaz que busca el hiperplano óptimo que separa las clases en el espacio de características.

Metodología

El proyecto se ejecutó siguiendo las fases de la metodología CRISP-DM, adaptadas a los requerimientos específicos del análisis de movimiento en video.

Comprensión del Negocio y los Datos

En esta fase inicial, se definieron los objetivos y los requisitos del sistema. El problema se enmarca como un desafío de clasificación de series temporales con restricciones de tiempo real. Se establecieron las cinco actividades a reconocer y se planificó la recolección de un dataset diverso, incluyendo grabaciones propias con múltiples voluntarios, ángulos de cámara y velocidades de ejecución para asegurar la generalización del modelo.

Preparación de Datos e Ingeniería de Características

Este fue un paso crucial implementado a través de los scripts data_processing.py y feature_engineering.py. El flujo de trabajo fue el siguiente:

1. **Extracción de Landmarks:** Se procesaron los videos de entrada (.mp4) para extraer, cuadro por cuadro, las coordenadas 3D (x, y, z) de 18 articulaciones clave usando MediaPipe Pose. Estos datos, junto con la etiqueta de la actividad y el número de fotograma, se almacenaron en un archivo raw_landmarks.csv.
2. **Cálculo de Características Derivadas:** A partir de los landmarks crudos, se calcularon características biomecánicas como el ángulo de la rodilla, la inclinación del torso y la velocidad de las extremidades. Estas nuevas características se añadieron a nuestro conjunto de datos para enriquecerlo.
3. **Limpieza y Normalización:** Los datos se limpiaron rellenando valores faltantes. Posteriormente, se aplicó StandardScaler para normalizar las características, asegurando que el modelo no fuera sesgado por diferencias en la escala de las coordenadas o la complexión física de los sujetos.
4. **Reducción de Dimensionalidad:** Se utilizó el Análisis de Componentes Principales (PCA) para reducir la dimensionalidad del vector de características, conservando el 95% de la varianza. Esto ayuda a agilizar el entrenamiento y la inferencia en tiempo real, además de

mitigar el riesgo de sobreajuste.

Modelado

Esta fase, implementada en el script train_model.py, se centró en entrenar el clasificador:

- **División de Datos:** El dataset procesado se dividió en conjuntos de entrenamiento (80%) y prueba (20%), utilizando una estratificación por clase para manejar cualquier desbalance en el número de ejemplos por actividad.
- **Entrenamiento del Clasificador:** Se entrenó un modelo Support Vector Machine (SVC) con un kernel Gausiano (RBF) y el parámetro probability=True activado, lo cual es necesario para obtener una puntuación de confianza en las predicciones.
- **Almacenamiento de Artefactos:** El modelo SVM entrenado, el codificador de etiquetas (LabelEncoder), el normalizador (StandardScaler) y el modelo PCA se guardaron como archivos .pkl para su uso posterior en la fase de despliegue.

Evaluación

Para medir el rendimiento del modelo, se definieron un conjunto de métricas clave que combinan la precisión de la clasificación con la eficiencia del sistema:

- Métricas de Clasificación: Precisión, Recall y F1-Score para evaluar el balance entre falsos positivos y negativos por cada clase.
- Métricas de Rendimiento: Latencia (tiempo de procesamiento por fotograma) y FPS (cuadros por segundo) para verificar que la solución cumple con los requisitos de tiempo real.

La evaluación formal se realiza sobre el conjunto de pruebas, que contiene datos no vistos por el modelo durante el entrenamiento.

Despliegue

La fase final consistió en integrar los componentes en una aplicación funcional de tiempo real, como se muestra en realtime_classifier.py:

1. **Lógica en Tiempo Real:** Se desarrolló una clase ActivityClassifier que carga los artefactos serializados (modelo SVM, PCA, Scaler y LabelEncoder).
2. **Procesamiento de Video en Vivo:** La aplicación captura video de la cámara y, para cada fotograma, aplica la secuencia completa: extracción de landmarks, construcción y transformación del vector de características, y predicción.
3. **Heurística y Suavizado de Predicciones:** Se implementaron dos mejoras clave para la estabilidad:
    a. **Detección de Inactividad:** Un umbral de movimiento total (total_movement) permite clasificar al usuario como "quieto" de forma heurística, sin sobrecargar al modelo SVM.
    b. **Suavizado Temporal:** Se utiliza un búfer (deque de tamaño 10) que almacena las probabilidades de las últimas predicciones. La clasificación final se basa en un promedio ponderado de este búfer, dando más importancia a los fotogramas recientes. Esto evita saltos abruptos en la clasificación.
4. **Interfaz Gráfica (GUI):** Utilizando OpenCV, se creó una ventana que muestra el video con el esqueleto superpuesto, la etiqueta de la actividad predicha, la confianza del modelo, un análisis postural con ángulos y un contador de FPS para monitorear el rendimiento.

Resultados

Los resultados cuantitativos del modelo SVM se obtuvieron tras la evaluación sobre el conjunto de prueba, que contenía 893 muestras no vistas durante el entrenamiento.

Métricas de Rendimiento del Clasificador

El modelo alcanzó una exactitud (accuracy) general del 94%. El rendimiento detallado para cada una de las ocho clases se presenta en la siguiente tabla:

Actividad	Precisión	Recall (Sensibilidad)	F1-Score	Soporte (Muestras)
adelante	0.9	0.82	0.86	148
atras	0.98	0.96	0.97	102
derecha	1	0.99	0.99	92
inclinarse	0.95	0.93	0.94	88
izquierda	1	0.99	0.99	84
pararse	0.97	0.98	0.98	119
retroceder	0.81	0.93	0.87	146
sentarse	0.99	0.97	0.98	114
Promedio Macro	0.95	0.95	0.95	893
Promedio Ponderado	0.94	0.94	0.94	893

Fuente: Reporte de clasificación generado por *train_model.py.*

Matriz de Confusión

Aunque no se incluye la visualización gráfica de la matriz de confusión, el reporte de clasificación anterior permite inferir las áreas de mayor y menor acierto del modelo, como se detalla en la sección de análisis.

Rendimiento en Tiempo Real

La validación cualitativa se realizó a través de una demostración en vivo del sistema. Se observó que la aplicación es capaz de clasificar las acciones de un usuario frente a la cámara con una latencia baja, permitiendo una interacción fluida. El contador de FPS integrado en la interfaz gráfica confirmó un rendimiento adecuado para la operación en tiempo real en el hardware de prueba.

Análisis de Resultados

El análisis de las métricas revela un modelo de alto rendimiento, pero con áreas claras de oportunidad.

- Rendimiento General Sólido: Una exactitud del 94% y un F1-Score promedio (macro y ponderado) de 0.95 y 0.94 respectivamente, indican que el modelo es robusto y está bien balanceado en general. La alta correlación entre el promedio macro y ponderado sugiere que el modelo no está sesgado hacia las clases con más muestras.
- Clases de Alto Desempeño: Actividades como derecha, izquierda, pararse, sentarse y atras muestran un F1-Score de 0.97 o superior. Esto demuestra que las características de ingeniería (ángulos, distancias y orientación) son altamente discriminatorias para estas acciones, que involucran cambios de postura y desplazamientos laterales bien definidos.
- Áreas de Confusión y Mejora: Las clases con menor rendimiento son adelante y retroceden.
    - adelante (F1-Score: 0.86): Su recall de 0.82 indica que el modelo no detectó el 18% de las veces que una persona caminaba hacia adelante, confundiéndose probablemente con otra acción similar como retroceder.
    - retroceder (F1-Score: 0.87): Su precisión de 0.81 significa que de todas las veces que el modelo predijo "retroceder", el 19% eran en realidad otra acción.
    - Hipótesis: La confusión entre avanzar y retroceder es lógica, ya que ambas son formas de caminar. La dificultad del modelo para distinguirlas sugiere que las características basadas en la pose de un solo fotograma no capturan completamente la dinámica de la

profundidad. Aunque PCA fue clave para la generalización, podría estar filtrando sutiles variaciones en el eje Z que distinguen estos dos movimientos.

Conclusiones y Trabajo Futuro

Conclusiones

En este proyecto, se desarrolló con éxito un sistema robusto de reconocimiento de actividades humanas, logrando una exactitud del 94% mediante el uso de PCA y un clasificador SVM. El proceso nos enseñó la importancia crítica de una ingeniería de características detalladas, el impacto positivo de la reducción de dimensionalidad en la prevención del sobreajuste y la necesidad de una evaluación basada en métricas rigurosas. La implementación de heurísticas como el suavizado temporal y la detección de inactividad demostró ser fundamental para la estabilidad en una aplicación de tiempo real.

Este proyecto demostró con éxito la viabilidad de crear una herramienta de bajo costo y no invasiva para el análisis del movimiento humano en tiempo real. La exactitud del 94% alcanzada no es solo una métrica técnica, sino que se traduce en un alto grado de fiabilidad que podría habilitar aplicaciones prácticas en entornos controlados, como guiar a pacientes de fisioterapia en la ejecución correcta de sus ejercicios o monitorear actividades básicas de personas mayores para detectar patrones de movilidad anómalos.

El principal aprendizaje desde la perspectiva de la aplicación es que mientras las posturas estáticas y los cambios posturales (como sentarse o pararse) son identificados con una precisión casi perfecta, los movimientos dinámicos complejos (la confusión entre caminar adelante y retroceder) requieren un análisis temporal más avanzado. Esta distinción es crucial para el desarrollo futuro: un sistema para rehabilitación de rodilla podría estar casi listo para un piloto, mientras que una aplicación para análisis de la marcha en atletas necesitaría la incorporación de modelos secuenciales.

En resumen, más que un ejercicio técnico, este trabajo sienta las bases para el desarrollo de sistemas de retroalimentación biomecánica accesibles, capaces de ofrecer valor tangible en áreas como la salud, el bienestar y el deporte, democratizando el acceso a tecnologías de análisis de movimiento que tradicionalmente requieren equipos especializados y costosos.

Trabajo Futuro

Para mejorar y expandir el sistema, se proponen las siguientes líneas de trabajo:

- Mejorar la Discriminación Direccional: Investigar y diseñar características que capturen mejor la dinámica del movimiento en el eje Z para resolver la confusión entre "caminar hacia adelante" y "retroceder".
- Incorporar Análisis Temporal con Deep Learning: Expandir el conjunto de datos y experimentar con modelos de redes neuronales recurrentes (LSTM) o Transformers, que están diseñados para analizar secuencias y podrían mejorar la precisión en acciones complejas.
- Despliegue en Dispositivos de Borde (Edge Devices): Optimizar el modelo (ej. usando cuantización para su ejecución en dispositivos con menor capacidad computacional, como una Raspberry Pi o Jetson Nano, ampliando sus posibles aplicaciones.
- Enriquecer la Interfaz: Mejorar la GUI para mostrar gráficos históricos de los ángulos y velocidades, ofreciendo una retroalimentación biomecánica más completa al usuario.

Referencias Bibliográficas

1. Google. (s.f.). MediaPipe Pose. Google for Developers. Recuperado de https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker
2. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0: Step-by-step data mining guide. SPSS Inc.