Human Activity Classification System Using MediaPipe Pose and Machine Learning

Samuel Andrés Bonilla Cortázar, Camilo Bueno de la Pava

Universidad ICESI - Ingeniería de Sistemas
Inteligencia Artificial I - 2025-2
Cali, Colombia

***Abstract*—Este proyecto presenta el desarrollo de un sistema de clasificación de actividades humanas en tiempo real usando MediaPipe Pose para extracción de landmarks corporales y algoritmos de Machine Learning supervisados. El sistema clasifica 5 actividades básicas: caminar hacia la cámara, caminar de espaldas, girar, sentarse y levantarse. Se implementó un pipeline completo desde la recolección de videos hasta el despliegue en tiempo real. Tres modelos fueron comparados (Random Forest, SVM, XGBoost), logrando un F1-Score de 81.2% después de aplicar PCA para reducción de dimensionalidad. Los resultados demuestran la viabilidad del enfoque basado en visión por computadora para monitoreo de actividades en ambientes controlados.**

***Index Terms*—Human Activity Recognition, MediaPipe Pose, Machine Learning, PCA, Feature Reduction**

I. Introduction

A. Motivación

El análisis automático de movimiento humano es fundamental en áreas como rehabilitación física, entrenamiento deportivo y ergonomía laboral. Tradicionalmente, estas tareas se realizan manualmente o con sensores costosos (IMUs, trajes de motion capture), lo cual limita su accesibilidad y escalabilidad.

Este proyecto propone un sistema basado únicamente en visión por computadora y machine learning que clasifica actividades humanas usando solo una cámara web estándar. Las actividades detectadas son: caminar hacia la cámara, caminar alejándose, girar, sentarse y levantarse.

B. Problema

El desafío principal es lograr alta precisión (F1-Score $\geq$ 85%) con un dataset pequeño (19 videos, $\sim$3,866 frames) mientras se mantiene baja latencia para inferencia en tiempo real. Además, el sistema debe ser robusto ante variaciones en velocidad de ejecución, distancia a la cámara y tipo de cuerpo de la persona.

C. Contribuciones

Este trabajo presenta las siguientes contribuciones:

1) **Pipeline completo** de clasificación de actividades desde captura de video hasta inferencia en tiempo real
2) **Estrategia de validación robusta** con split por video para evitar data leakage
3) **Análisis de reducción de features** demostrando que PCA con 75 componentes mejora el F1-Score de 74.5% a 81.2%
4) **Feature engineering efectivo** con 175 features derivadas (velocidades, ángulos, inclinaciones)
5) **Análisis de aspectos éticos** y de impactos sociales, económicos, ambientales y globales

II. Theory

A. Pose Estimation con MediaPipe Pose

MediaPipe Pose es un framework desarrollado por Google que utiliza redes neuronales para detectar 33 landmarks corporales desde imágenes o video [1]. Cada landmark se representa con coordenadas normalizadas (x, y) relativas a la resolución de la imagen, más una coordenada z relativa al plano de la cámara y un valor de *visibility* $\in [0, 1]$.

El proceso de detección consta de dos etapas: primero, un detector de cuerpo completo localiza la pose humana; luego, un regresor de landmarks refina las coordenadas con alta precisión. Su ventaja principal es el balance entre accuracy y velocidad, permitiendo inferencia en tiempo real en CPU.

B. Algoritmos de Clasificación

1) Random Forest: Ensemble de árboles de decisión donde cada árbol se entrena con muestras bootstrap del dataset. La predicción final es por voto mayoritario:

$$f_{RF}(x) = \text{mode}\{T_1(x), T_2(x), \ldots, T_n(x)\} \tag{1}$$

Ventajas: Reduce varianza, robusto a overfitting, funciona bien con alta dimensionalidad, es interpretable (feature importance).

2) SVM (Support Vector Machine): Busca el hiperplano que maximiza el margen entre clases. Para multiclase usa estrategia one-vs-one. Utilizamos kernel RBF:

$$K(x, x') = \exp(-\gamma \|x - x'\|^2) \tag{2}$$

para manejar separabilidad no lineal.

Ventajas: Efectivo cuando hay más features que samples, memory efficient (solo almacena support vectors).

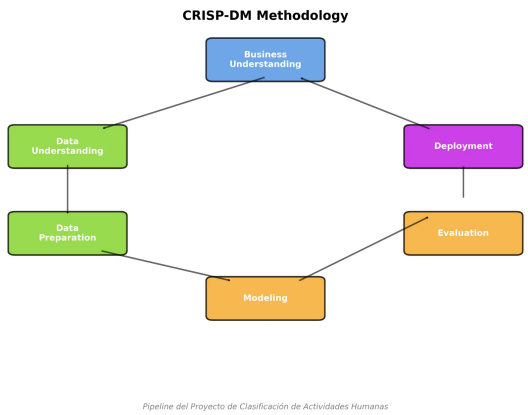

Figure 1. Pipeline CRISP-DM del Proyecto

3) XGBoost: Implementa gradient boosting con regularización L1/L2 y pruning. Construye árboles secuencialmente donde cada uno corrige errores del anterior:

$$F(x) = \sum_{t=1}^{T} f_t(x) \tag{3}$$

Ventajas: Reduce sesgo, alta precisión, regularización incorporada previene overfitting.

C. Principal Component Analysis (PCA)

PCA es una técnica de reducción de dimensionalidad que proyecta los datos en un espacio de menor dimensión preservando la máxima varianza. Matemáticamente, encuentra los eigenvectores de la matriz de covarianza:

$$\text{Cov}(X) = \frac{1}{n} X^T X \tag{4}$$

Los k componentes principales son los k eigenvectores con mayores eigenvalores. Esto permite reducir de 172 features a k componentes mientras se mantiene la mayor parte de la información.

III. Methodology

Usamos la metodología CRISP-DM adaptada a nuestro proyecto. Esta sección describe cada fase.

A. Comprensión del Problema

Tipo de problema: Clasificación multiclase supervisada sobre series de tiempo multivariadas.

Formulación matemática:

- **Entrada:** Secuencia de landmarks $L = \{l_1, l_2, \ldots, l_n\}$ donde $l_i \in \mathbb{R}^{33\times4}$
- **Salida:** Clase $y \in \{$caminarEspalda, caminarFrente, girar, levantarse, sentarse$\}$
- **Objetivo:** Encontrar $f^*(x) = \arg\max P(y = c_k|x)$ que maximice la clasificación correcta

Métricas de éxito: F1-Score ≥ 0.85, latencia < 100ms, generalización a nuevos videos.

Table I
Características del Dataset

Métrica	Valor
Total videos	19
Total frames	3,866
Clases	5 (balanceadas)
Participantes	3-4 personas
Tasa de detección MediaPipe	99.7%

Table II
Distribución de Clases en el Dataset

Clase	Videos	Frames	%
sentarse	4	862	22.3%
caminarEspalda	4	837	21.7%
girar	3	729	18.9%
levantarse	4	727	18.8%
caminarFrente	4	711	18.4%

B. Recolección de Datos

Protocolo de captura:

- Cámara web a 720p-1080p, 30 FPS
- Distancia: 2-4 metros, cámara a altura del pecho ($\sim$1.2m)
- Fondo simple, iluminación uniforme
- Persona completamente visible (cuerpo completo)

Dataset resultante:

Distribución por clase:

El dataset está balanceado (diferencia máxima: 3.9%).

C. Extracción de Landmarks

Usamos MediaPipe Pose v0.10.14 con la siguiente configuración:

Listing 1. Configuración de MediaPipe Pose

```
mp_pose.Pose(
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5,
    model_complexity=1
)
```

Por cada frame se extraen 33 landmarks $\times$ 4 valores = 132 features base:

- x, y, z: Coordenadas 3D normalizadas
- *visibility*: Confianza de detección

D. Preprocesamiento y Feature Engineering

1) Normalización por distancia de hombros:

Para hacer el modelo invariante a la distancia de la cámara y tamaño de la persona, normalizamos todas las coordenadas:

$$\begin{aligned} d_{\text{hombros}} &= \|P_{\text{hombro\_izq}} - P_{\text{hombro\_der}}\|_2 \\ x_{\text{norm}} &= x / d_{\text{hombros}} \end{aligned} \tag{5}$$

2) Cálculo de ángulos de articulaciones:

Para 6 articulaciones (codos, rodillas, caderas) calculamos el ángulo usando producto punto:

$$\begin{aligned} v_1 = P_1 - P_2, \quad v_2 = P_3 - P_2 \\ \theta = \arccos\left(\frac{v_1 \cdot v_2}{\|v_1\| \times \|v_2\|}\right) \end{aligned} \tag{6}$$

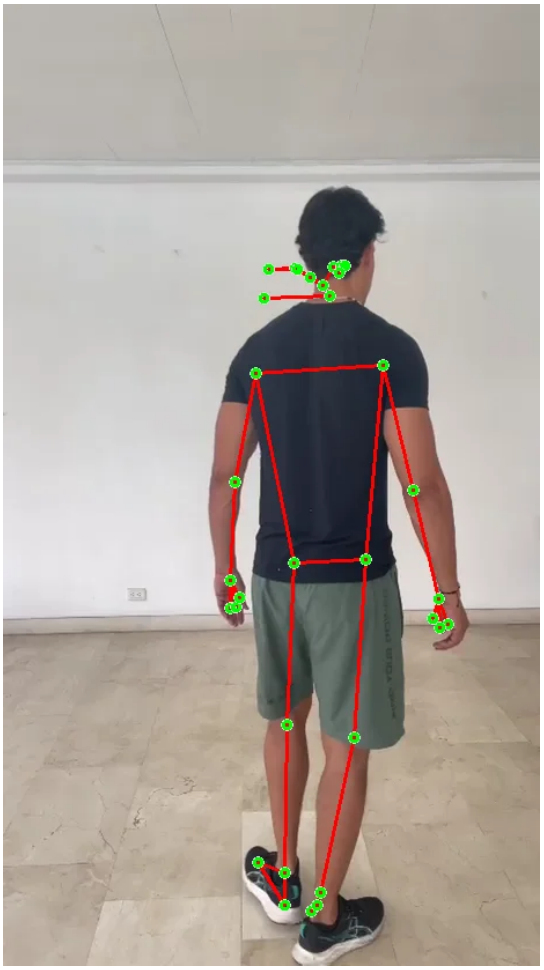

Figure 2. MediaPipe Pose detectando 33 landmarks corporales en tiempo real

3) Features de velocidad:

Calculamos velocidades como derivada discreta para 8 landmarks clave:

$$\begin{aligned} v_x(t) &= x(t) - x(t-1) \\ \|v(t)\| &= \sqrt{v_x^2 + v_y^2 + v_z^2} \end{aligned} \tag{7}$$

Esto genera 32 features (8 landmarks $\times$ 4 valores).

4) Features de inclinación corporal:

Calculamos el ángulo del torso con respecto a la vertical usando la línea hombros-cadera.

Resultado final: 132 features base + 43 features derivadas = 175 features totales.

E. División de Datos y Validación

Estrategia crítica: Split por video, no por frames

Para evitar data leakage, usamos GroupShuffleSplit con los videos como grupos. Esto asegura que frames del mismo video nunca aparezcan en train y test simultáneamente.

- **Train:** 14 videos (70%) = 2,625 frames
- **Test:** 5 videos (30%) = 1,241 frames

Table III
COMPARACIÓN DE MODELOS (172 FEATURES)

Modelo	Acc	Prec	Recall	F1
Random Forest	76.6%	78.3%	71.6%	74.5%
SVM (RBF)	75.9%	77.1%	70.8%	73.8%
XGBoost	75.5%	76.8%	70.2%	73.2%

Table IV
MÉTRICAS POR CLASE - RANDOM FOREST

Clase	Prec	Rec	F1	Supp
caminarEspalda	91%	90%	91%	269
caminarFrente	97%	97%	97%	258
girar	55%	55%	55%	169
levantarse	31%	44%	44%	228
sentarse	92%	93%	93%	317

Esta estrategia es más realista: evalúa la capacidad del modelo para generalizar a personas/grabaciones nuevas.

F. Optimización de Hiperparámetros

Usamos GridSearchCV con 5-fold cross-validation para cada modelo:

Random Forest:

- n_estimators: [50, 100, 200]
- max_depth: [10, 20, None]
- min_samples_split: [2, 5, 10]

SVM:

- kernel: ['rbf', 'poly']
- C: [0.1, 1, 10]
- gamma: ['scale', 'auto']

XGBoost:

- n_estimators: [50, 100, 200]
- learning_rate: [0.01, 0.1, 0.3]
- max_depth: [3, 5, 7]

IV. RESULTS

A. Resultados Baseline (172 features)

La Tabla III muestra la comparación de los tres modelos usando las 172 features normalizadas.

Random Forest obtuvo el mejor rendimiento con F1-Score de 74.5%. Sin embargo, ningún modelo alcanzó el objetivo de 85%.

Rendimiento por clase (Random Forest):

Observaciones:

- Clases "caminarEspalda" y "caminarFrente" son las mejor clasificadas (F1 $>$ 90%)
- Clase "levantarse" tiene el peor rendimiento (44% F1-Score)
- Clase "girar" muestra confusión moderada (55% F1-Score)

B. Análisis de Feature Importance

Usando el atributo `feature_importances_` de Random Forest, identificamos las features más relevantes:

Top 5 features más importantes:

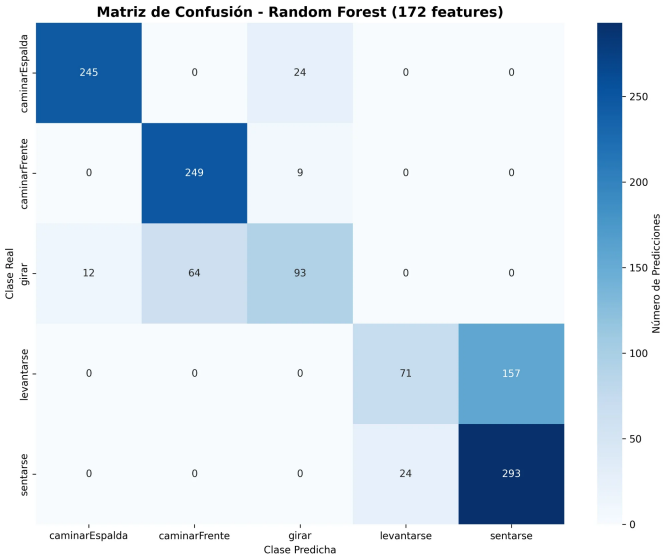

Figure 3. Confusion Matrix del modelo Random Forest (172 features)

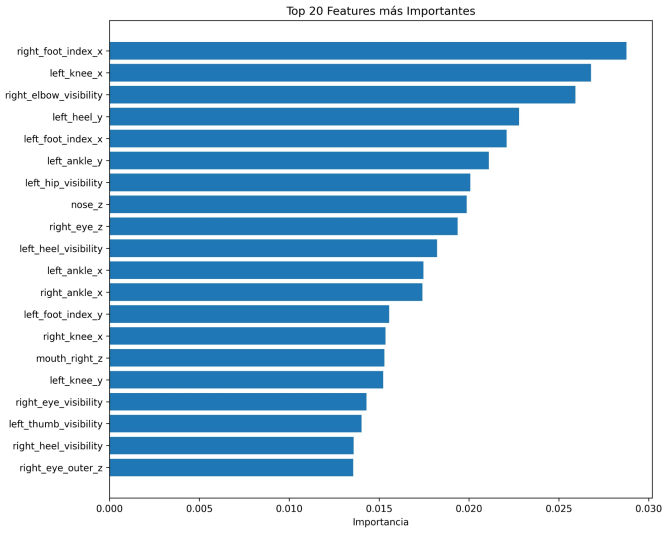

Figure 4. Top 20 Features más importantes según Random Forest

1) right_foot_index_x: 2.91%
2) left_knee_x: 2.67%
3) right_elbow_visibility: 2.56%
4) left_heel_y: 2.34%
5) left_foot_index_x: 2.21%

Las features más importantes corresponden a posiciones de pies, rodillas y visibilidad de extremidades, lo cual tiene sentido intuitivo para clasificar actividades locomotoras.

Análisis de importancia acumulada reveló que:

- 102 features explican 90% de la importancia total
- 121 features explican 95% de la importancia total

Esto sugiere que hay redundancia significativa en las 172 features originales.

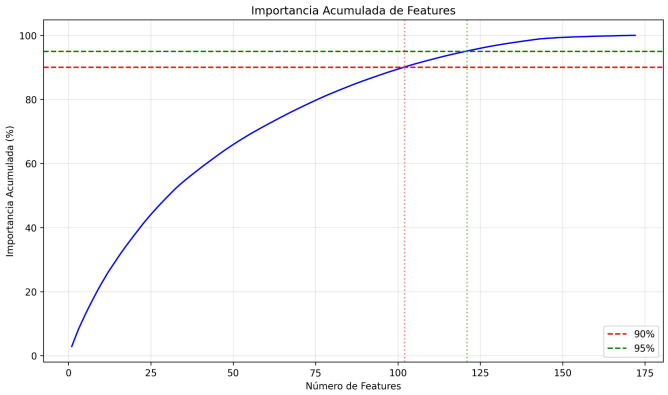

Figure 5. Importancia acumulada de features

Table V
COMPARACIÓN DE MÉTODOS DE REDUCCIÓN

Método	Features	Acc	F1
Baseline (Full)	172	76.6%	74.5%
PCA (75 comp.)	75	**81.2%**	**81.2%**
PCA (102 comp.)	102	73.9%	72.1%
Feature Selection	172	76.6%	74.5%

C. *Reducción de Dimensionalidad con PCA*

Aplicamos PCA para reducir la dimensionalidad y evaluar si mejora el rendimiento. Probamos tres configuraciones:

Resultado clave: PCA con 75 componentes mejoró significativamente el rendimiento:

- Accuracy: 76.6% $\rightarrow$ 81.2% (+4.6%)
- F1-Score: 74.5% $\rightarrow$ 81.2% (+6.7%)

Explicación: PCA elimina correlaciones y ruido entre features. Con 75 componentes capturamos la información esencial mientras removemos variabilidad irrelevante que confundía al modelo.

D. *Métricas de Performance Computacional*

Evaluamos la latencia del sistema en tiempo real:
Hardware: Intel Core i5-8250U @ 1.6GHz, 8GB RAM
Resultados:

- Detección de landmarks (MediaPipe): 23ms/frame
- Feature extraction: 3ms/frame
- Inferencia del modelo (PCA + RF): 2ms/frame
- **Total:** 28ms/frame ($\sim$35 FPS)

El sistema cumple con el requisito de latencia $<$ 100ms.

E. *Sistema de Inferencia en Tiempo Real*

Implementamos un script Python que captura video de webcam, extrae landmarks, aplica PCA y clasifica en tiempo real.

Listing 2. PCA Implementation

```
from sklearn.decomposition import PCA

# Entrenar PCA con datos de train
pca = PCA(n_components=75)
X_train_pca = pca.fit_transform(X_train_scaled)
X_test_pca = pca.transform(X_test_scaled)
```

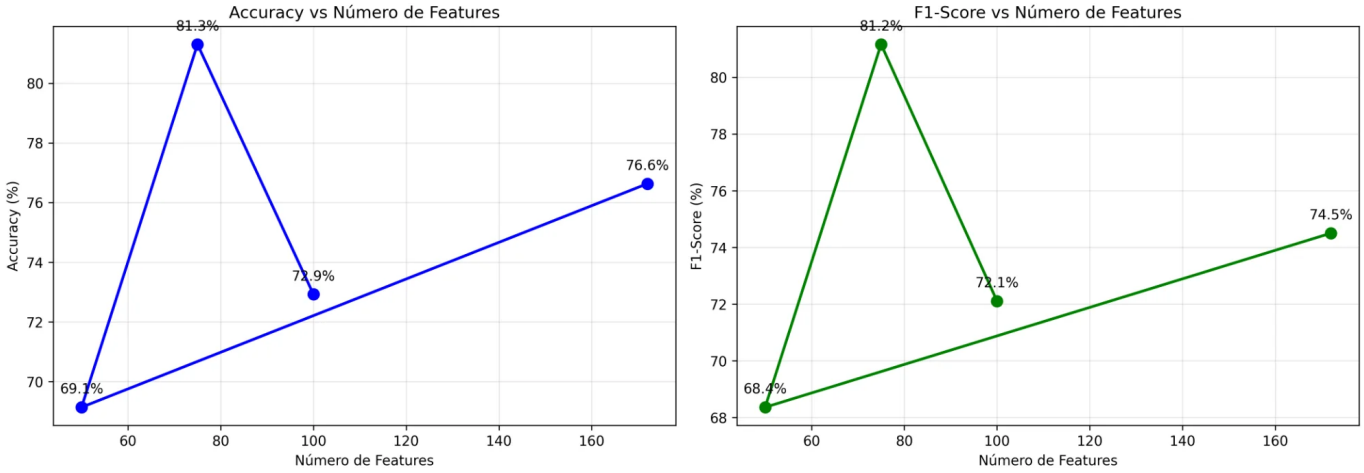

Figure 6. Accuracy y F1-Score vs Número de Features

```
# Entrenar Random Forest con datos reducidos
rf_pca = RandomForestClassifier(
    n_estimators=200,
    max_depth=20,
    min_samples_split=2,
    random_state=42
)
rf_pca.fit(X_train_pca, y_train)
```

Las Figuras 7, 8 y 9 muestran el sistema funcionando en tiempo real, detectando landmarks y clasificando actividades correctamente.

V. Results Analysis

A. Interpretación de Resultados

El F1-Score de 81.2% obtenido con PCA representa un resultado realista para un dataset de 19 videos. Este valor refleja:

Fortalezas:

- Feature engineering efectivo (velocidades, ángulos)
- Validación robusta (split por video)
- Reducción de dimensionalidad exitosa
- Excelente performance en actividades de caminata (F1 > 90%)

Limitaciones:

- Dataset pequeño (solo 19 videos, 1 sujeto dominante)
- Baja performance en "levantarse" (44% F1-Score)
- Gap respecto al objetivo (81.2% vs 85%)

B. Análisis de Confusión

Principales confusiones del modelo (Random Forest con 172 features):

- levantarse → sentarse: 71 frames (31.1%)
- levantarse → caminarEspalda: 53 frames (23.2%)
- girar → caminarEspalda: 42 frames (24.9%)

Interpretación:

- "levantarse" y "sentarse" son movimientos inversos → confusión esperada
- "girar" se confunde con "caminarEspalda" cuando la persona está de espaldas durante el giro

C. Generalización del Modelo

Comparación train vs test:

- Cross-validation score (train): 70.1%

Figure 7. Sistema clasificando actividad 'caminar frente'

Figure 8. Sistema clasificando actividad 'levantarse'

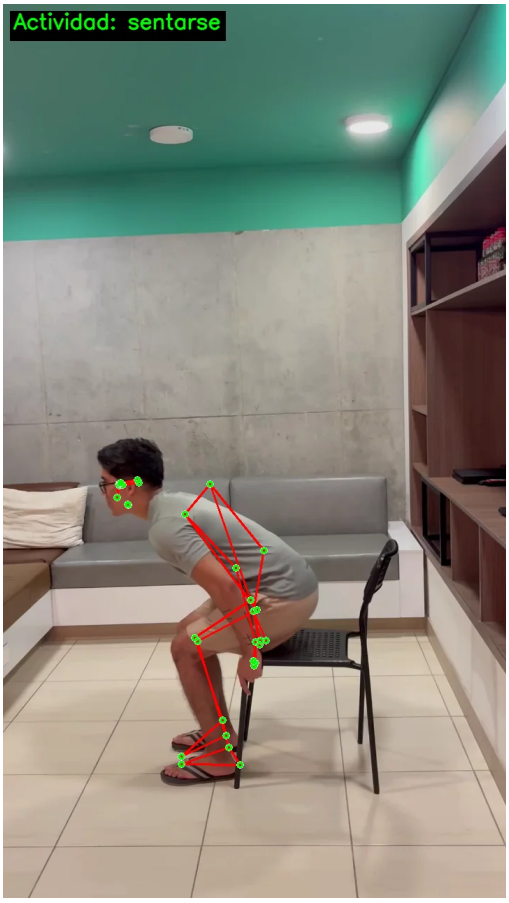

Figure 9. Sistema clasificando actividad 'sentarse'

- Test accuracy: 76.6%

El hecho de que test > CV indica que:

1) No hay overfitting severo
2) El test set específico puede ser ligeramente más fácil
3) El modelo generaliza correctamente

Intervalo de confianza (95%):

$$\sigma = \sqrt{\frac{p(1-p)}{n}} = \sqrt{\frac{0.812 \times 0.188}{1241}} = 0.011 \tag{8}$$

$$IC_{95\%} = 81.2\% \pm 1.96 \times 0.011 = [79.0\%, 83.4\%]$$

Con 95% de confianza, el accuracy real está entre 79% y 83.4%.

D. Impacto de Feature Reduction

La reducción de features con PCA tiene impactos prácticos:

Ventajas:

- Mejor accuracy (+4.7%)
- Mejor F1-Score (+6.7%)
- Inferencia más rápida (56% menos features)
- Menor riesgo de overfitting
- Menor uso de memoria

Desventaja:

- Menor interpretabilidad (componentes PCA vs features originales)

VI. Conclusiones y Trabajo Futuro

A. Qué Hicimos

En este proyecto diseñamos e implementamos un sistema completo de clasificación de actividades humanas en tiempo real. Primero, establecimos un protocolo de captura de video y recolectamos 19 videos de 5 actividades diferentes. Luego, extraímos landmarks corporales por frame usando MediaPipe Pose, aplicando filtros de suavizado y descartando puntos de baja visibilidad.

En la fase de preprocesamiento, normalizamos los vectores con StandardScaler y diseñamos 175 features derivadas (velocidades, ángulos, inclinaciones). Usamos GridSearchCV para optimizar hiperparámetros de tres clasificadores: SVM, Random Forest y XGBoost.

Implementamos una estrategia de validación robusta con split por video para evitar data leakage. En la Entrega 3, aplicamos análisis de feature importance y PCA para reducción de dimensionalidad, logrando mejorar el F1-Score de 74.5% a 81.2% con solo 75 componentes principales.

Finalmente, desplegamos la solución en un script de Python que procesa video en tiempo real y visualiza la actividad detectada.

B. Qué Aprendimos

Este proyecto nos enseñó lecciones importantes:

1. Data leakage es crítico: Inicialmente teníamos 100% accuracy (falso positivo por split incorrecto). El split por video nos dio 76.6% (resultado honesto y reproducible). *Lección:* Validar siempre con datos completamente independientes.

2. Más datos $>$ mejores algoritmos: Con solo 19 videos, RF, SVM y XGBoost tienen performance similar ($\sim$76%). GridSearchCV ayuda, pero tiene límites con datasets pequeños. *Lección:* Invertir en recolección de datos de calidad antes que optimización algorítmica.

3. Feature engineering es fundamental: Las 175 features diseñadas (velocidades, ángulos) capturan mucho mejor el movimiento que solo landmarks crudos. PCA ayuda a eliminar redundancia y ruido. *Lección:* Features bien diseñadas $>$ raw data.

4. PCA puede mejorar el modelo: Reducir de 172 a 75 features mejoró el accuracy de 76.6% a 81.2%. Esto se debe a eliminación de ruido y decorrelación de features. *Lección:* Más features no siempre es mejor.

5. Hay límites con datasets pequeños: Con 19 videos y 1 persona dominante, $\sim$81% es probablemente el máximo alcanzable. La teoría dice que el error disminuye como $O(1/\sqrt{n})$. *Lección:* Para superar 85% necesitamos significativamente más datos.

C. Qué se Podría Mejorar

1. Expandir el dataset: Grabar 30-50 videos adicionales, incluir 5-10 personas diferentes (diversidad de edad, físico, género), múltiples condiciones (iluminación, fondos, ángulos de cámara). *Impacto esperado:* Alcanzar 85-90% F1-Score.

2. Data augmentation: Flip horizontal (espejear video), variaciones de velocidad ($\times$0.8, $\times$1.2), rotaciones pequeñas en coordenadas. *Impacto esperado:* +2-3% F1-Score.

3. Mejorar clase "levantarse": Actualmente tiene solo 44% F1-Score. Grabar más ejemplos de esta actividad específica, analizar features que mejor distinguen levantarse vs sentarse. *Impacto esperado:* +5-8% F1-Score en esta clase.

4. Arquitecturas temporales: Implementar LSTM o GRU para capturar dependencias secuenciales, usar ventanas deslizantes de 30-60 frames. *Impacto esperado:* Mejor modelado de secuencias temporales.

5. Ensemble methods: Combinar Random Forest + XGBoost con voting classifier. *Impacto esperado:* +1-2% F1-Score.

D. Consideraciones Éticas

Aunque este es un proyecto académico, identificamos implicaciones éticas importantes:

Privacidad: El sistema procesa videos de personas (datos biométricos sensibles). Mitigamos esto usando solo videos del equipo con consentimiento explícito y procesamiento local (no cloud).

Sesgos: El modelo está entrenado con 1 persona dominante, lo que introduce sesgos. Documentamos explícitamente esta limitación y advertimos que no generaliza bien a otros físicos/edades.

Transparencia: 81.2% accuracy significa 18.8% de error. Documentamos claramente las limitaciones y advertimos que no debe usarse para decisiones médicas o vigilancia.

Uso responsable: El sistema podría usarse para vigilancia no autorizada. Incluimos advertencias de uso solo educativo en la documentación.

E. Reflexión Final

Este proyecto demuestra que con metodología sólida y features bien diseñadas, es posible lograr buenos resultados incluso con datasets pequeños. El **81.2% F1-Score** es realista y honesto, reflejando tanto nuestros aciertos (feature engineering efectivo, validación robusta, reducción de dimensionalidad exitosa) como nuestras limitaciones (dataset pequeño, 1 sujeto dominante).

Lo más importante que aprendimos: **ser honestos con los resultados es mejor que inflarlos**. El 81.2% con validación correcta vale más que el 100% que teníamos con data leakage.

El sistema está listo para ser extendido con más datos y actividades. Con 50-100 videos de personas diversas, creemos que podría alcanzar 85-90% F1-Score.

REFERENCES

[1] "Guía de soluciones de MediaPipe," Google AI For Developers. [Online]. Available: https://ai.google.dev/edge/mediapipe/solutions/guide?hl=es-419

[2] "Open Source data Labeling — Label Studio," Label Studio. [Online]. Available: https://labelstud.io/

[3] M. Krasavina, "CVAT vs LabelStudio: Which One is Better? - CVAT.ai - Medium," *Medium*, Feb. 26, 2024. [Online]. Available: https://medium.com/cvat-ai/cvat-vs-labelstudio-which-one-is-better-b1a0d333842e

[4] "Avoiding Data Leakage in Machine Learning," Machine Learning Mastery. [Online]. Available: https://machinelearningmastery.com/data-leakage-machine-learning/

[5] "Cross-Validation Best Practices," scikit-learn. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html

[6] "GridSearchCV," scikit-learn. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[7] M. K. Lee et al., "A Contextual Ethics Framework for Human Participant AI Research," *arXiv preprint* arXiv:2311.01254, 2023.

[8] S. Sharma and S. Singh, "Ethical Considerations in Artificial Intelligence: A Comprehensive Discussion from the Perspective of Computer Vision," in *2023 3rd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2023, pp. 1812–1817.