



Universidad Icesi

CLASIFICADOR DE MOVIMIENTOS

Autores:

Juan Felipe Jojoa
Felipe Rojas

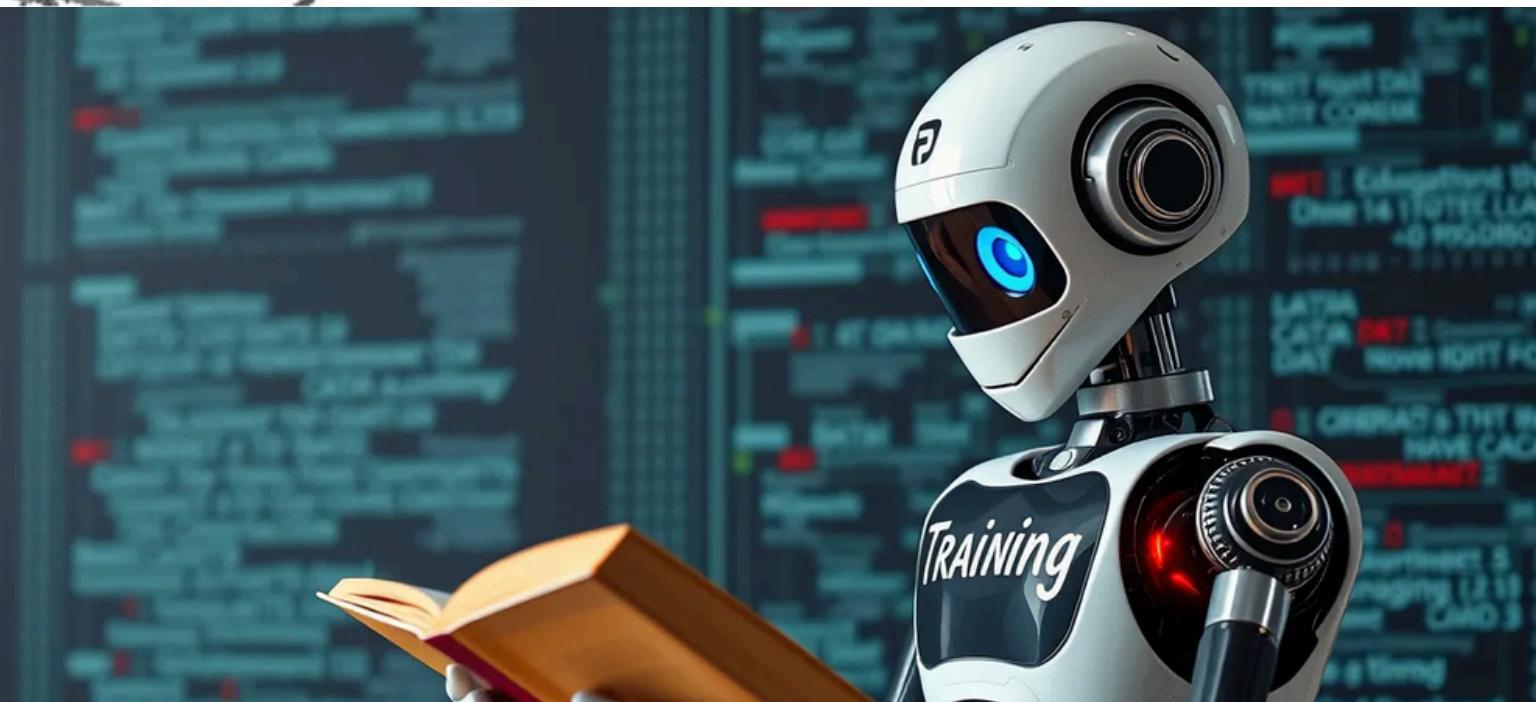
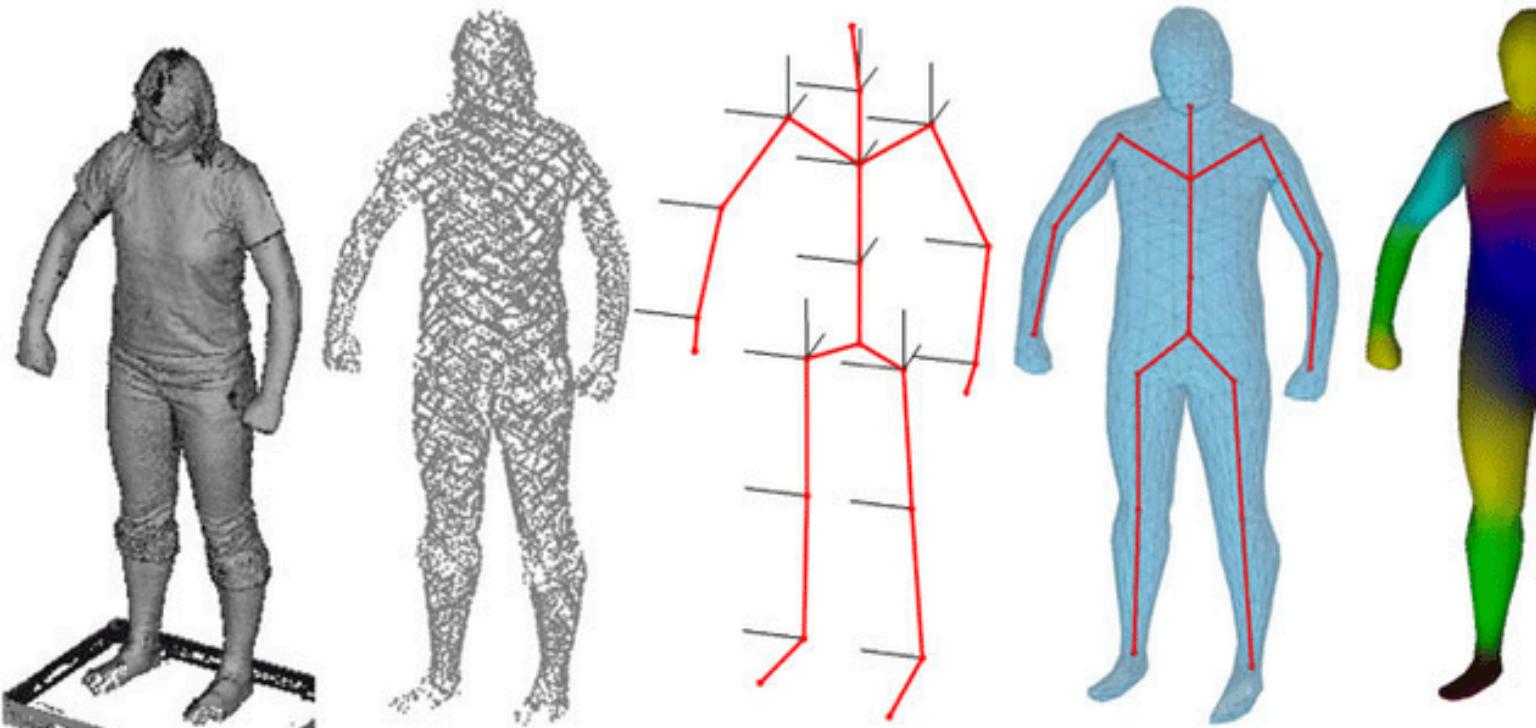
INTRODUCCIÓN Y PROBLEMÁTICA

El reconocimiento automático de actividades humanas representa uno de los desafíos más relevantes en visión por computadora y machine learning, con aplicaciones que van desde el monitoreo médico hasta el análisis deportivo.

- Identificación automática de actividades específicas a partir de video
- Extracción de patrones biomecánicos significativos
- Clasificación precisa en tiempo real



Schicht	Datum	Borts	05.01.01
V-Achse		FLO	
Stoffwechsel DIP / ESK		U637	
V-Sch		U63	
V-Papierrolle		mmsys	
Arbeitsbreite		mmms	
Stoffaufstand		m	
Austauschrahmen			
Druck	mmlys	1.018	
PD Innendruck	mmlys	1.0019	
Druckwaage / Sollung	mm	2	
Lippenöffnung	mm	1.3	
Vorwand	mm	1.3	
Fuß - Stoffauspumpe	mm	1.20	
Sch - za Freq.	mm	94.5	
Jack Hub	mm	2.80	
Deformer D	mm	2.50	
Oberstecknässerung			
Summer / Einwasserung in	% / Wm	1.3	
1. Zone	% / mm	1.2	
Druck Leiste 1 + 2	% / mm	1.2	
Druck Leiste 2	mbar	20	
Druck Leiste 4	mbar	20	
Druck Leiste 5 + 6	mbar	1.50	
Druck Leiste 7 + 8	mbar	1.50	
Druck Leiste 9 + 10	mbar	1.50	
Deformer D / Spalt	mm	0.682	
Vakuum / Entnahmestellung		1.10	
1. Vakuums	mbar	-1.2	
Vakuum / Entnahmest	mbar	-1.2	
Doppelzähne	mbar	-1.3	
Summer	mbar	-1.0	
1. Formationszone / Zone	mbar	-1.10	
2. Zone / 1. Trocken	mbar	-1.20	
Trockensauger	mbar	-1.20	
Flechsauger	mbar	-1.20	
SSW	mbar	-1.20	
PU Hölzerzone	mbar	-1.20	
PU Preßzone	mbar	-1.20	
Pressenpartie / Linie	mbar	-1.20	
1 Press	mbar	-1.20	



MARCO TEÓRICO

Normalización de Datos

- Reducción de dependencia del ángulo de grabación
- Alineación respecto al centro de las caderas
- Mejora de consistencia entre frames y sujetos

Biomecánica Articular

- Velocidades articulares
- Ángulos en rodillas y articulaciones clave
- Distancia entre hombros
- Deltas de movimiento entre frames consecutivos

Modelos Supervisados

- Random Forest (RF): Ensamble de árboles de decisión
- XGBoost: Gradient boosting con árboles secuenciales

HERRAMIENTAS Y TECNOLOGÍAS

Extracción de Características

- MediaPipe: Detección de landmarks corporales
- OpenCV: Procesamiento de video

Machine Learning

- Random Forest: Clasificación robusta
- XGBoost: Optimización de gradiente
- Scikit-learn: Pipeline de ML

Optimización

- GridSearchCV: Búsqueda exhaustiva de hiperparámetros
- Optuna: Optimización bayesiana

Análisis de Datos

- Pandas: Manipulación de datos
- Matplotlib/Seaborn: Visualización

ENTRENAMIENTO DEL MODELO

Fase 1: Entrenamiento Inicial

- Script: Entrenamiento_CPU.py
- Dataset: outputRecords.csv
- Optimización: GridSearchCV con validación cruzada
- Limitación: Desbalance de clases afectó el rendimiento

Fase 2: Optimización Avanzada

- Script: EntrenamientoBayesianNetwork.py
- Dataset: outputRecords_balanceado.csv
- Optimización: Optuna con optimización bayesiana
- Validación cruzada estratificada (3 folds)
- Mejora significativa en precisión y recall

Modelos Finales:

- trainedModel.pkl (Fase 1)
- best_pose_classifier.pkl (Fase 2)

```
optimize_hyperparameters(self, X_train_scaled, y_train):
    """
    Encuentra los mejores hiperparámetros para Random Forest y XGBoost usando GridSearchCV.
    La selección del mejor *tipo* de modelo (RF o XGB) se basa en su rendimiento de CV en el conjunto de entrenamiento.
    Args:
        X_train_scaled: Datos de entrenamiento normalizados.
        y_train: Etiquetas de entrenamiento.
    Returns:
        dict: Diccionario con los mejores parámetros para cada modelo y el modelo recomendado.
    """
    logging.info("Iniciando optimización de hiperparámetros...")

    # Definición del grid de hiperparámetros para Random Forest.
    rf_param_grid = {
        'n_estimators': [50, 100, 200],
        'max_depth': [None, 10, 20, 30],
        'min_samples_split': [2, 5, 10],
        'min_samples_leaf': [1, 2, 4],
        'class_weight': ['balanced', 'balanced_subsample', None] # Para manejar desbalance si existe
    }

    # Definición del grid de hiperparámetros para XGBoost.
    xgb_param_grid = {
        'max_depth': [3, 5, 7],
        'learning_rate': [0.01, 0.1, 0.2],
        'n_estimators': [100, 200, 300],
        'min_child_weight': [1, 3, 5],
        'gamma': [0, 0.1, 0.2],
        'subsample': [0.7, 0.8, 0.9],
        'colsample_bytree': [0.7, 0.8, 0.9] # Muestreo de columnas
    }

    # Estrategia de validación cruzada (estratificada para clasificación)
    cv_strategy = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

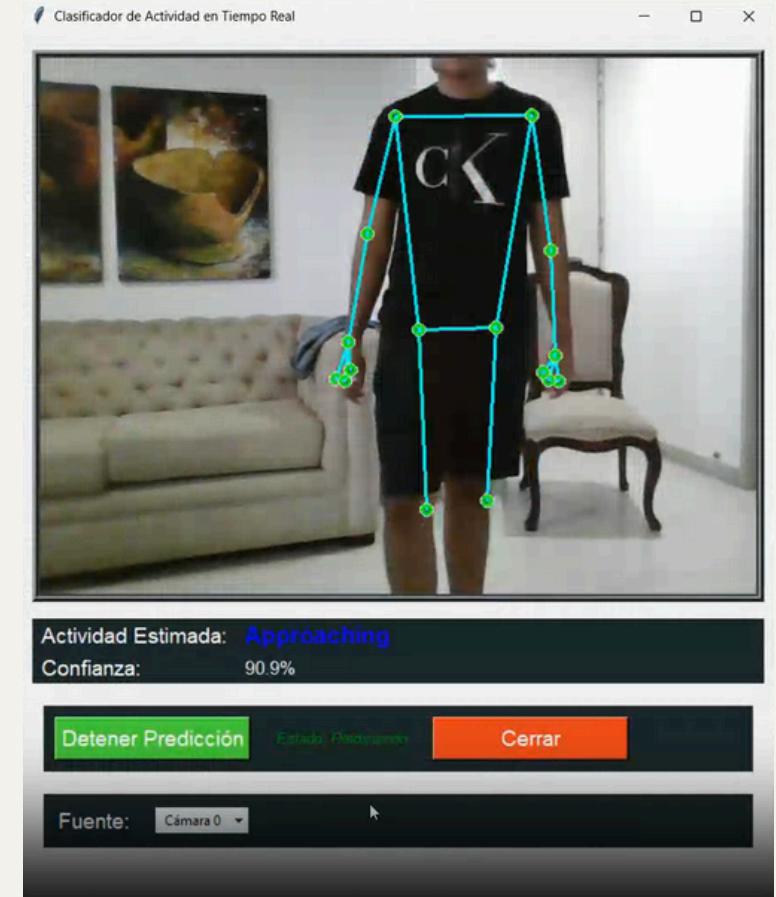
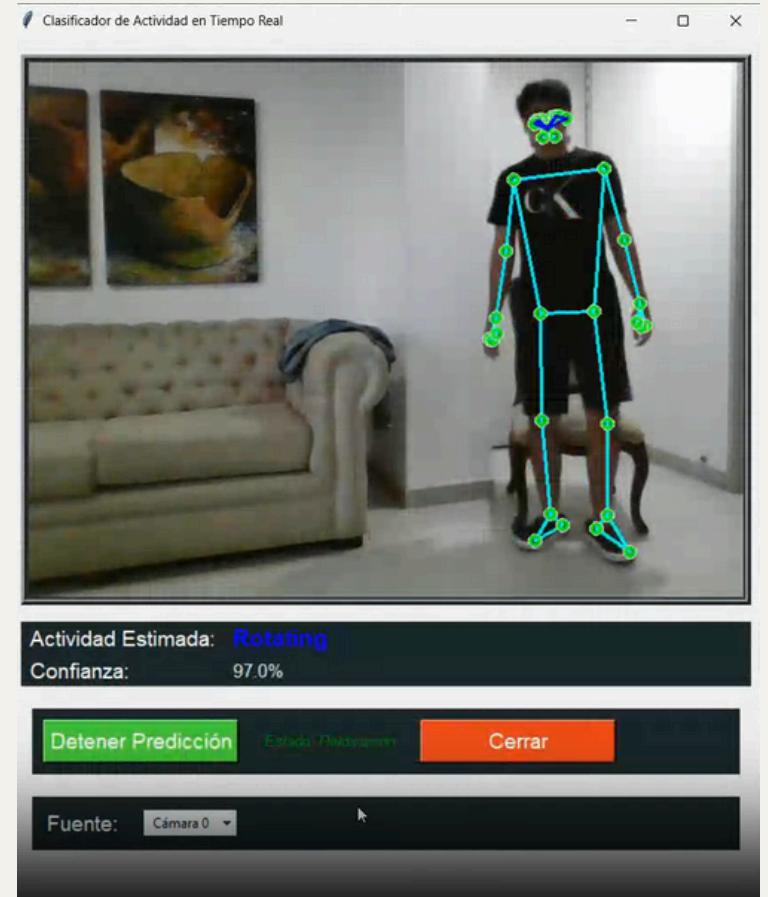
    # Optimización de Random Forest
    rf = RandomForestClassifier(random_state=42)
    rf_grid = GridSearchCV(estimator=rf, param_grid=rf_param_grid, cv=cv_strategy, n_jobs=-1, verbose=1, scoring='accuracy')
    rf_grid.fit(X_train_scaled, y_train)
    logging.info(f"Mejores parámetros para Random Forest: {rf_grid.best_params_}")
    logging.info(f"Mejor puntuación CV para Random Forest: {rf_grid.best_score_:.4f}")

    # Optimización de XGBoost
    xgb_model = xgb.XGBClassifier(random_state=42, use_label_encoder=False, eval_metric='mlogloss')
    xgb_grid = GridSearchCV(estimator=xgb_model, param_grid=xgb_param_grid, cv=cv_strategy, n_jobs=-1, verbose=1, scoring='accu
```

RESULTADOS - EVOLUCIÓN DEL MODELO

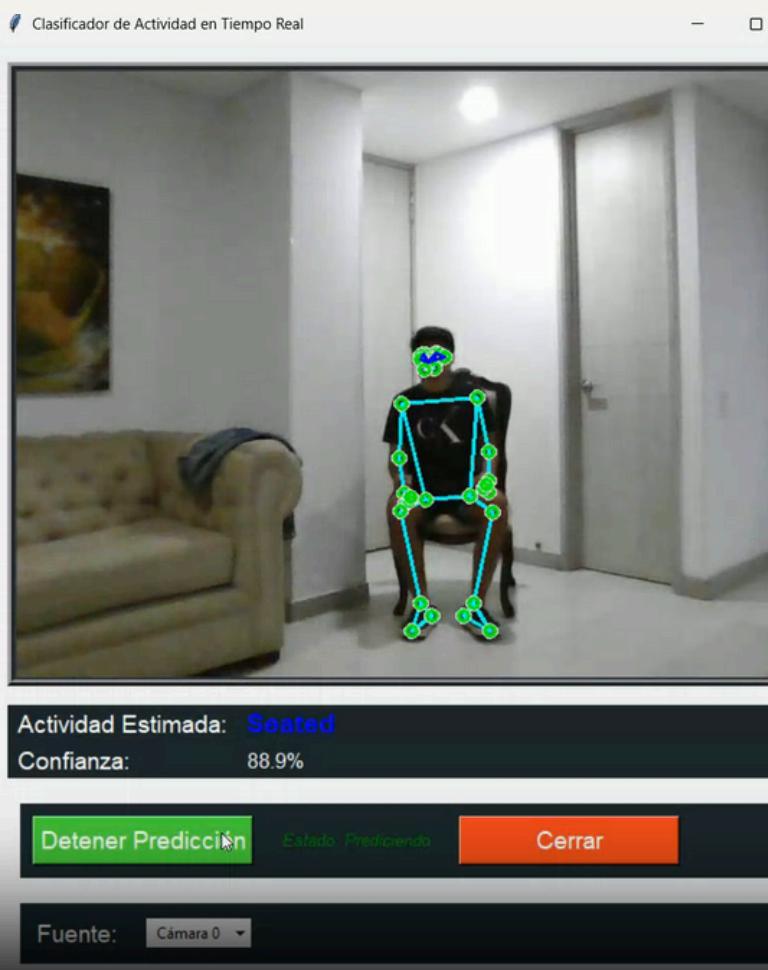
Modelo con Coordenadas Absolutas

- Utilizó directamente coordenadas (x, y, z)
- Problema: Overfitting severo
- Causa: Aprendía posiciones específicas en la imagen
- Resultado: Poca generalización en condiciones reales



Modelo con Características Derivadas

- Utilizó métricas biomecánicas calculadas
- Ventaja: Independiente de posición en imagen
- Resultado: Mayor robustez y generalización
- Captura patrones universales de movimiento humano



ANÁLISIS DE RENDIMIENTO



OPTUNA

Desafíos Identificados:

- Dificultad para distinguir acciones similares:
 - "Acercarse" vs "Caminar"
 - "Estar de pie" vs "Levantarse"
- Necesidad de métricas más discriminativas
- Importancia del dataset balanceado

Herramientas de Análisis:

- Visualizaciones de Optuna
- Gráficos de distribución de clases
- Análisis de características más relevantes

```
--- Métricas de Evaluación en Conjunto de Prueba ---
```

```
2025-06-12 18:08:51,221 - INFO - Accuracy: 0.9774
```

```
2025-06-12 18:08:51,221 - INFO -
```

```
Informe de Clasificación Detallado:
```

	precision	recall	f1-score	support
approaching	1.00	0.98	0.99	52
departing	1.00	0.97	0.98	63
rotating	1.00	1.00	1.00	56
seated	0.94	0.94	0.94	51
upright	0.94	1.00	0.97	44
accuracy			0.98	266
macro avg	0.98	0.98	0.98	266
weighted avg	0.98	0.98	0.98	266

ANÁLISIS DE RENDIMIENTO RNN

Desafíos Identificados:

- Necesidad de métricas más discriminativas
- Importancia del dataset balanceado y una mayor cantidad de datos para el enfoque usado

Herramientas de Análisis:

- Gráficos de distribución de clases
- Análisis de características más relevantes

		precision	recall	f1-score	support
	approaching	1.0	0.9807692307692307	0.9902912621359223	52.0
	departing	0.984375	1.0	0.9921259842519685	63.0
	rotating	1.0	1.0	1.0	56.0
	seated	0.9607843137254902	0.9607843137254902	0.9607843137254902	51.0
	upright	0.9545454545454546	0.9545454545454546	0.9545454545454546	44.0
	accuracy	0.981203007518797	0.981203007518797	0.981203007518797	0.981203007518797
	macro avg	0.9799409536541891	0.9792197998080351	0.9795494029317672	266.0
	weighted avg	0.9812617481203008	0.981203007518797	0.9811995587930151	266.0

CONCLUSIONES

Logros Principales

- Sistema funcional de análisis de actividades humanas
- Integración completa del pipeline de ML
- Interfaz visual interactiva para evaluación en tiempo real

Contribuciones Técnicas

- Metodología robusta de extracción de características biomecánicas
- Comparativa entre enfoques de coordenadas vs métricas derivadas
- Implementación de optimización bayesiana para hiperparámetros
- Solución al problema de desbalance de clases

Valor del Proyecto

- Demostración exitosa de técnicas avanzadas de ML
- Base sólida para sistemas más complejos de análisis de movimiento
- Contribución al campo de Computer Vision y ML aplicado
- Validación de metodologías innovadoras en reconocimiento de patrones

BIBLIOGRAFÍA

- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system
- Breiman, L. (2001). Random forests. Machine Learning
- Akiba, T., et al. (2019). Optuna: A next-generation hyperparameter optimization framework
- Google. (2024). MediaPipe: Cross-platform, customizable ML solutions
- Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python



Universidad Icesi

MUCHAS GRACIAS

Agosto 2025