Integrantes:

* Sebastian Hidalgo
* Ricardo Medina
* Alejandro Osejo

# Proyecto de Sistema de Anotación de Vídeo

## Abstract

## Introduction

Este proyecto tiene como objetivo desarrollar un software especializado en la anotación de vídeos que capture diversas actividades humanas, como caminar, girar, sentarse y estar de pie. Para lograr esto, se implementará un sistema de análisis de vídeo en tiempo real, lo que permitirá la detección y seguimiento preciso de los movimientos articulares y de la postura de los individuos en las grabaciones. La solución se enfocará en proporcionar herramientas efectivas para la clasificación y análisis de estas actividades, contribuyendo así a campos como la investigación en comportamiento humano, la rehabilitación física y la interacción hombre-máquina.

## Theory

1. Preprocesamiento y Escalado de Datos: Pandas y NumPy son fundamentales para cargar, limpiar y manipular datos en estructuras eficientes. El escalado mediante MinMaxScaler y StandardScaler estandariza las características, lo cual es esencial en modelos sensibles a magnitudes, como KNN y redes neuronales.
2. División de Datos: Dividir los datos en conjuntos de entrenamiento y prueba con train\_test\_split permite evaluar si el modelo generaliza bien en datos no vistos.
3. Modelos de Clasificación: Se utilizan cuatro modelos:
   * KNeighborsClassifier (KNN): Clasificación basada en la proximidad.
   * RandomForestClassifier: Modelo de ensamble robusto frente al sobreajuste.
   * SVC: Clasificación mediante hiperplanos en espacios de alta dimensionalidad.
   * MLPClassifier: Red neuronal para patrones complejos y no lineales.
4. Optimización de Parámetros: GridSearchCV ayuda a encontrar la mejor combinación de parámetros para el modelo, utilizando validación cruzada.
5. Métricas de Evaluación: classification\_report y accuracy\_score brindan medidas detalladas como precisión y exhaustividad para evaluar el rendimiento del modelo en cada clase.
6. Visualización de Datos: Seaborn y Matplotlib permiten visualizar patrones y relaciones en los datos, ayudando en el análisis y selección de modelos.

## Methodology

* Recolección y Preprocesamiento de Datos: Recopilamos videos que muestran actividades humanas clave como caminar y sentarse. Realizamos anotaciones manuales iniciales y limpiamos los datos con Pandas y NumPy. Usamos MinMaxScaler y StandardScaler para estandarizar las características, lo cual es fundamental para mejorar la precisión de los modelos.
* División de los Datos: Dividimos los datos en conjuntos de entrenamiento y prueba con train\_test\_split, asegurando una distribución equilibrada para evitar sesgos en la clasificación de actividades.
* Modelos de Clasificación: Implementamos cuatro modelos principales: KNeighborsClassifier para detección basada en proximidad, RandomForestClassifier por su resistencia al sobreajuste, SVC para clasificación en espacios complejos y MLPClassifier para patrones no lineales.
* Optimización de Parámetros y Evaluación: Utilizamos GridSearchCV para ajustar parámetros y classification\_report para evaluar precisión y exhaustividad en cada clase. La visualización con Seaborn y Matplotlib ayudó a identificar patrones y ajustar el modelo final.
* Implementación en Tiempo Real: Finalmente, desarrollamos una interfaz para anotación automática en tiempo real, usando el modelo mejor ajustado para la detección y clasificación de actividades en los videos.

## Results

## Results Analysis

## Conclusion and future works

## Bibliographic References

* Scikit-learn, “scikit-learn: machine learning in Python,” *Scikit-learn.org*, 2019. https://scikit-learn.org/stable/
* “Matplotlib documentation — Matplotlib 3.5.0 documentation,” *matplotlib.org*. https://matplotlib.org/stable/
* M. Waskom, “seaborn: statistical data visualization — seaborn 0.10.1 documentation,” *seaborn.pydata.org*, 2021. https://seaborn.pydata.org/index.html
* NumPy, “Overview — NumPy v1.19 Manual,” *numpy.org*, 2022. https://numpy.org/doc/stable/
* “pandas documentation — pandas 1.0.3 documentation,” *pandas.pydata.org*. https://pandas.pydata.org/docs/index.html