Universidad ICESI | Facultad Barberi de Ingeniería, Diseño y Ciencias Aplicadas

JUNE 2025

Real-Time Human Activity Detection and Classification System

Final Project - Artificial Intelligence I

PRESENTED TO

Milton Orlando Sarria Paja

PRESENTED BY

Andrés Pino

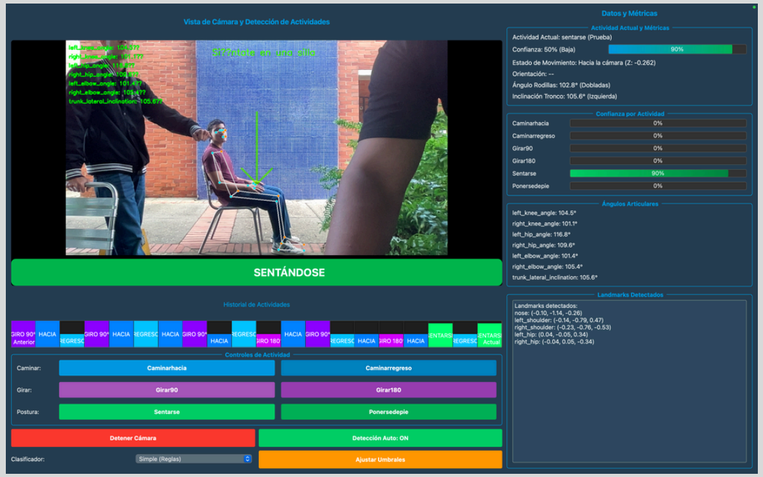
Jhonatan Castaño

What is this project?

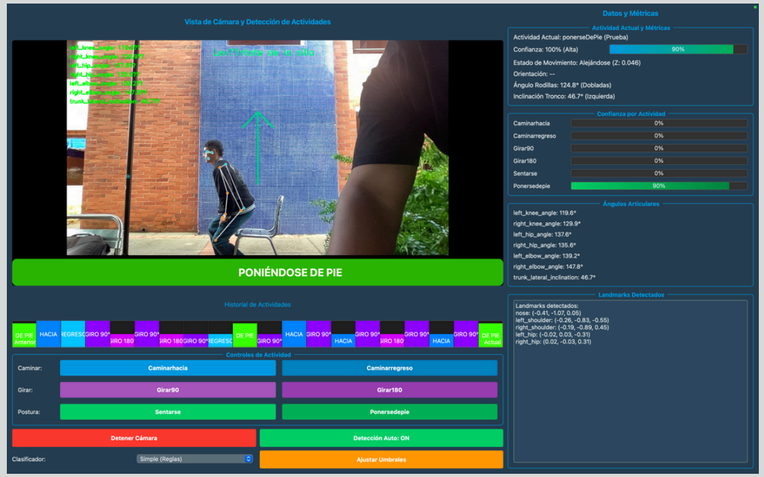
Teaching a Machine to Understand Human Action

We have built a system that uses a standard webcam to identify 6 basic human activities in real time.

One of the most representative screenshots of the system in action



Sitting Down



Standing up

Our Goal: The 6 core Activites

- Walking Towards 
- Walking Away 
- Turning 90° 
- Turning 180° 
- Sitting Down 
- Standing Up 

System Architecture: How Does It Work?

Our Methodology: From Pixels to Predictions → A 4-step workflow to process video and classify the activity

Pose Detection With MediaPipe

We use Google's MediaPipe to detect 33 key 3D body landmarks (X, Y, Z).

Feature Extraction

We convert the landmarks into 101 numerical features to describe the action.

Model Training

We trained and compared 6 different classification model.

The System in Action

Images in the next page →

PoseDetection With MediaPipe

The foundational step: where the system converts raw video pixels into structured information.

1. A single frame is captured from the video stream (either from a file or a live webcam).

2. This frame is fed into Google’s MediaPipe Pose model.

3. MediaPipe uses a pre-trained ML model to detect and return 3D coordinates (x, y, z) and a visibility score for 33 specific landmarks on the human body.

PoseDetection With MediaPipe

These are the standalone Python scripts:

- **extract_video_features.py:** This is the first step.
- **generate_report_plots.py:** To understand the data that we've just extracted before training a model.
- **train_classifier.py:** This is the core of the machine leaning process.
- **run_pipeline.py:** This script is an orchestrator.

System Architecture: How Does It Work?

Our Methodology: From Pixels to Predictions → A 4-step workflow to process video and classify the activity

Pose Detection With MediaPipe

We use Google's MediaPipe to detect 33 key 3D body landmarks (X, Y, Z).

Feature Extraction

We convert the landmarks into 101 numerical features to describe the action.

Model Training

We trained and compared 6 different classification model.

The System in Action

Images in the next page →

Feature Extraction

How does it work?

Geometric Features: This step takes the raw 3D coordinates of the 33 landmarks between joints. For example, by using the coordinates of the shoulder, elbow and wrist landmarks, it calculates the elbow angle. Similarly, it computes angles for the knees, hips, and shoulders.

Relational Features: It calculates distances and proportions, like the width between the shoulders or the hips.

Temporal (Dynamic) Features: The system compares a landmark's position in the current frame to its position in the previous frame to calculate its velocity. This captures the dynamics of the movement.

System Architecture: How Does It Work?

Our Methodology: From Pixels to Predictions → A 4-step workflow to process video and classify the activity

Pose Detection With MediaPipe

We use Google's MediaPipe to detect 33 key 3D body landmarks (X, Y, Z).

Feature Extraction

We convert the landmarks into 101 numerical features to describe the action.

Model Training

We trained and compared 6 different classification model.

The System in Action

Images in the next page →

Model Training

The “learning” phase

1. **Data Collection:** This create a large dataset where each row is a 101-feature vector, and each row is labeled with the correct activity (e.g., “sitting_down).
2. **Training:** The labeled dataset is fed into various classification algorithms (like logistic Regression, SVM, Random Forest). Each algorithm analyzes the data and learns the statistical patterns.
3. **Evaluation & Selection:** After training, the models are tested on a portion of the data they’ve never seen before. The model that performs the best is selected and saved to a file.

System Architecture: How Does It Work?

Our Methodology: From Pixels to Predictions → A 4-step workflow to process video and classify the activity

Pose Detection With MediaPipe

We use Google's MediaPipe to detect 33 key 3D body landmarks (X, Y, Z).

Feature Extraction

We convert the landmarks into 101 numerical features to describe the action.

Model Training

We trained and compared 6 different classification model.

The System in Action

Images of the system.

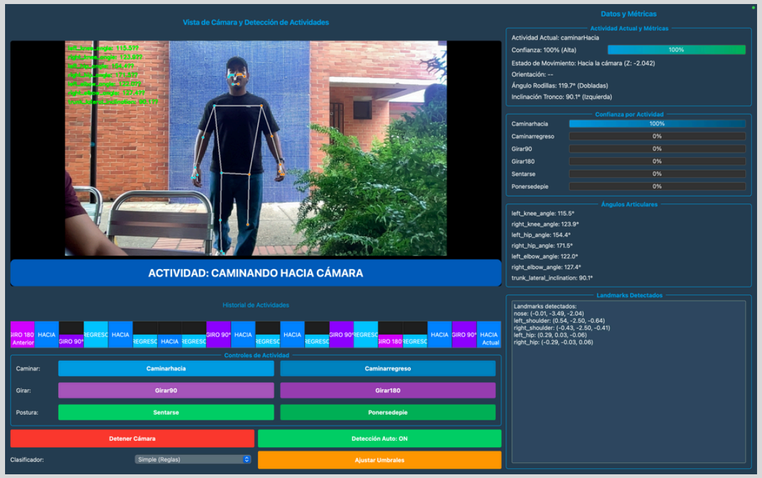
The System in Action

The live application where the trained model is used to make predictions on a real-time video from a webcam.

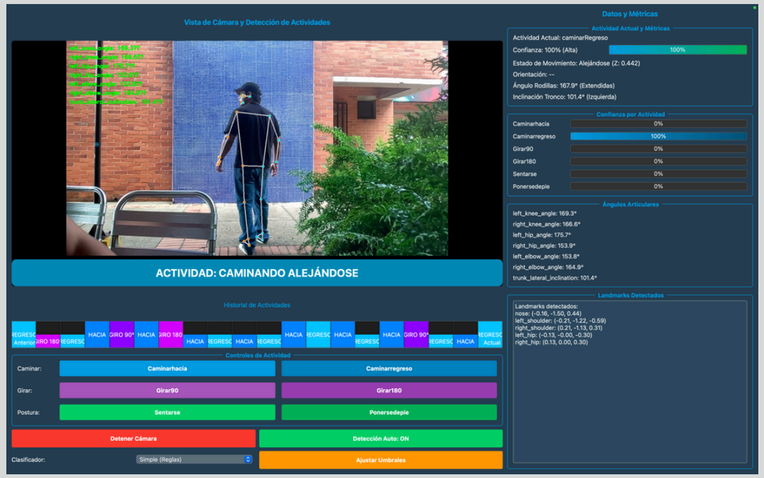
1. **Capture:** It grabs a new frame from the webcam.
2. **Detect & Extract:** Instantly performs Step 1 and Step 2 on that single frame to generate a 101-feature vector.
3. **Predict**: This new, inlabeled feature vector is fed into the loaded Logistic Regression Model (from Step 3).
4. **Display:** The model outputs a prediction. The application’s GUI then displays this predicted activity on the screen, and immediately gets ready for the next frame.

The System in Action

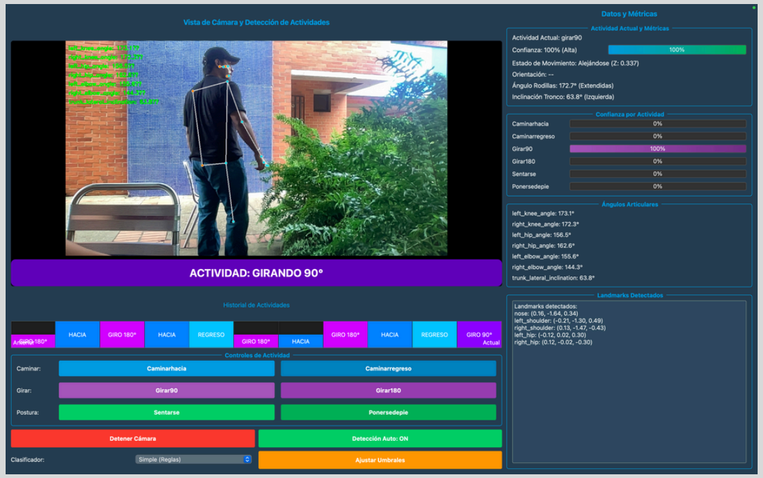
Walking Towards



Walking Away

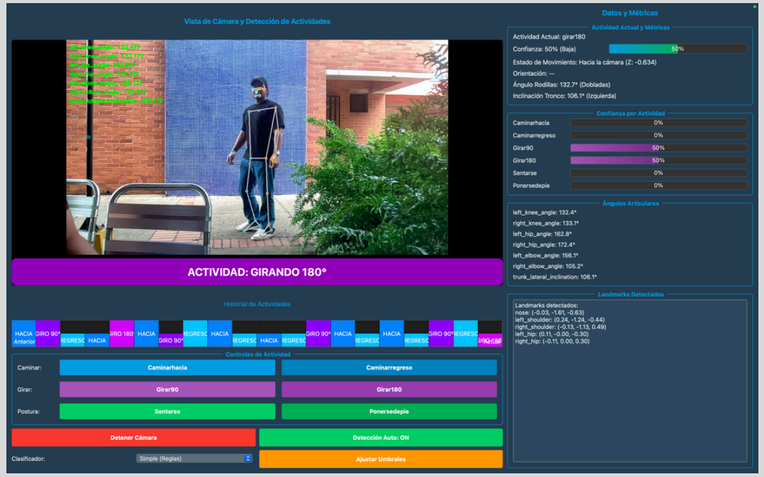


Turning 90°

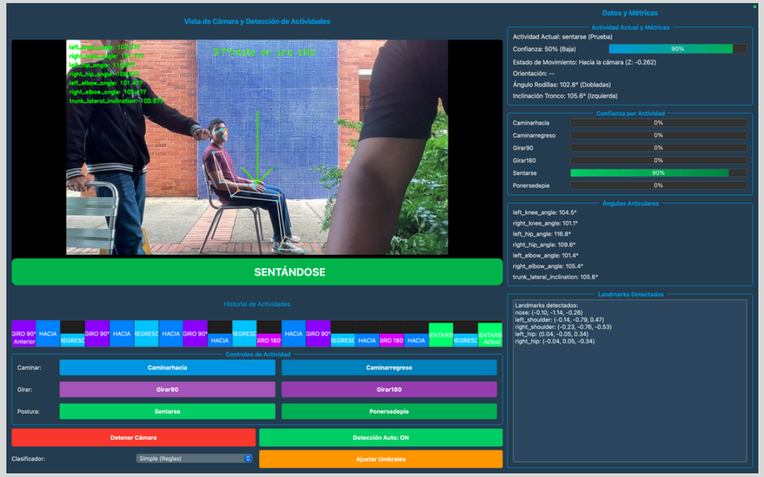


The System in Action

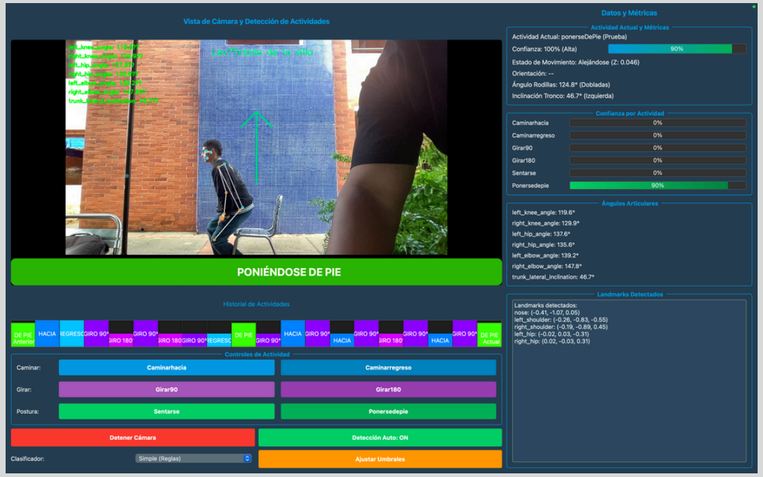
Turning 180°



Sitting Down



Standing Up



Results: Model Comparison

Modelo	Accuracy	Precision	Recall	F1-Score
RandomForest	0.6541	0.6563	0.6541	0.6551
SVM	0.6649	0.6599	0.6649	0.6603
LogisticRegression	**0.6946**	**0.6930**	**0.6946**	**0.6933**
GradientBoosting	0.6757	0.6757	0.6757	0.6757
KNN	0.6649	0.6564	0.6649	0.6598
XGBoost	0.6703	0.6705	0.6703	0.6703

The Logistic Regression model outperformed the others across key metrics

- **Mejor Modelo:** `LogisticRegression`
  - Accuracy: ~69.46%
  - Precision: ~69.30%
  - Recall: ~69.46%
  - F1-Score: ~69.33%

Main Achievements

- **End-to-End System:** From video capture to final prediction.
- **Practical Application of AI:** Real-world use of Computer Vision and ML techniques.
- **Interactive GUI:** A functional graphical interface for live demonstration.
- **Automated Pipeline:** Scripts to facilitate reproducibility and experimentation.

Conclusions & Future Work

Conclusions:

- Successfully built a functional, real-time HAR system.
- Logistic Regression proved to be the most effective model for our dataset.

Future Work:

- Expand the dataset (more data, more variety).
- Explore Deep Learning models (LSTMs, Transformers).
- Add more activities to recognize.

Thank You!