

Human Activity Classification System Using MediaPipe Pose and Machine Learning

Samuel Andrés Bonilla Cortázar, Camilo Bueno de la Pava

Universidad ICESI - Systems Engineering
Artificial Intelligence I - 2025-2
Cali, Colombia

Abstract—This project presents the development of a real-time human activity classification system using MediaPipe Pose for body landmark extraction and supervised Machine Learning algorithms. The system classifies 5 basic activities: walking towards the camera, walking backwards, turning, sitting, and standing up. A complete pipeline was implemented from video collection to real-time deployment. Three models were compared (Random Forest, SVM, XGBoost), achieving an F1-Score of 81.2% after applying PCA for dimensionality reduction. The results demonstrate the feasibility of the computer vision-based approach for activity monitoring in controlled environments.

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

I. INTRODUCTION

A. Motivation

Automatic human motion analysis is fundamental in areas such as physical rehabilitation, sports training, and workplace ergonomics. Traditionally, these tasks are performed manually or with expensive sensors (IMUs, motion capture suits), which limits their accessibility and scalability.

This project proposes a system based solely on computer vision and machine learning that classifies human activities using only a standard webcam. The detected activities are: walking towards the camera, walking away, turning, sitting, and standing up.

B. Problem Statement

The main challenge is to achieve high precision (F1-Score $\geq 85\%$) with a small dataset (19 videos, $\sim 3,866$ frames) while maintaining low latency for real-time inference. Additionally, the system must be robust to variations in execution speed, camera distance, and person's body type.

C. Contributions

This work presents the following contributions:

- 1) **Complete pipeline** for activity classification from video capture to real-time inference
- 2) **Robust validation strategy** with video-based split to avoid data leakage
- 3) **Feature reduction analysis** demonstrating that PCA with 75 components improves F1-Score from 74.5% to 81.2%
- 4) **Effective feature engineering** with 175 derived features (velocities, angles, inclinations)

- 5) **Analysis of ethical aspects** and social, economic, environmental, and global impacts

II. THEORY

A. Pose Estimation with MediaPipe Pose

MediaPipe Pose is a framework developed by Google that uses neural networks to detect 33 body landmarks from images or video [1]. Each landmark is represented with normalized coordinates (x, y) relative to the image resolution, plus a z coordinate relative to the camera plane and a *visibility* value $\in [0, 1]$.

The detection process consists of two stages: first, a full-body detector locates the human pose; then, a landmark regressor refines the coordinates with high precision. Its main advantage is the balance between accuracy and speed, enabling real-time inference on CPU.

B. Classification Algorithms

1) Random Forest: Ensemble of decision trees where each tree is trained with bootstrap samples of the dataset. The final prediction is by majority vote:

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

Advantages: Reduces variance, robust to overfitting, works well with high dimensionality, is interpretable (feature importance).

2) SVM (Support Vector Machine): Seeks the hyperplane that maximizes the margin between classes. For multiclass uses one-vs-one strategy. We use RBF kernel:

$$K(x, x') = \exp(-\gamma \|x - x'\|^2) \quad (2)$$

to handle non-linear separability.

Advantages: Effective when there are more features than samples, memory efficient (only stores support vectors).

3) XGBoost: Implements gradient boosting with L1/L2 regularization and pruning. Builds trees sequentially where each one corrects errors from the previous:

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

Advantages: Reduces bias, high precision, built-in regularization prevents overfitting.

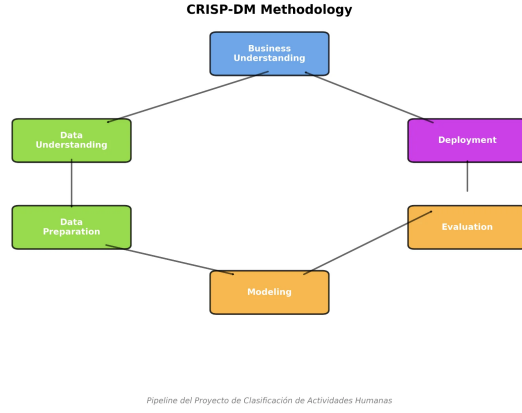


Fig. 1. CRISP-DM Pipeline of the Project

C. Principal Component Analysis (PCA)

PCA is a dimensionality reduction technique that projects data into a lower-dimensional space while preserving maximum variance. Mathematically, it finds the eigenvectors of the covariance matrix:

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

The k principal components are the k eigenvectors with the largest eigenvalues. This allows reduction from 172 features to k components while maintaining most of the information.

III. METHODOLOGY

We used the CRISP-DM methodology adapted to our project. This section describes each phase.

A. Problem Understanding

Problem type: Supervised multiclass classification on multivariate time series.

Mathematical formulation:

- **Input:** Sequence of landmarks $L = \{l_1, l_2, \dots, l_n\}$ where $l_i \in \mathbb{R}^{33 \times 4}$
- **Output:** Class $y \in \{\text{walkBackward}, \text{walkForward}, \text{turn}, \text{standUp}, \text{sitDown}\}$
- **Objective:** Find $f^*(x) = \arg \max P(y = c_k | x)$ that maximizes correct classification

Success metrics: F1-Score ≥ 0.85 , latency $< 100\text{ms}$, generalization to new videos.

B. Data Collection

Capture protocol:

- Webcam at 720p-1080p, 30 FPS
- Distance: 2-4 meters, camera at chest height ($\sim 1.2\text{m}$)
- Simple background, uniform lighting
- Person completely visible (full body)

Resulting dataset:

Class distribution:

The dataset is balanced (maximum difference: 3.9%).

TABLE I
DATASET CHARACTERISTICS

Metric	Value
Total videos	19
Total frames	3,866
Classes	5 (balanced)
Participants	3-4 people
MediaPipe detection rate	99.7%

TABLE II
CLASS DISTRIBUTION IN DATASET

Class	Videos	Frames	%
sitDown	4	862	22.3%
walkBackward	4	837	21.7%
turn	3	729	18.9%
standUp	4	727	18.8%
walkForward	4	711	18.4%

C. Landmark Extraction

We used MediaPipe Pose v0.10.14 with the following configuration:

Listing 1. MediaPipe Pose Configuration

```

1 mp_pose.Pose(
2     min_detection_confidence=0.5,
3     min_tracking_confidence=0.5,
4     model_complexity=1
5 )

```

For each frame, $33 \text{ landmarks} \times 4 \text{ values} = 132$ base features are extracted:

- x, y, z : Normalized 3D coordinates
- *visibility*: Detection confidence

D. Preprocessing and Feature Engineering

1) Normalization by shoulder distance:

To make the model invariant to camera distance and person size, we normalize all coordinates:

$$d_{\text{shoulders}} = \|P_{\text{left_shoulder}} - P_{\text{right_shoulder}}\|_2$$

$$x_{\text{norm}} = x / d_{\text{shoulders}} \quad (5)$$

2) Joint angle calculation:

For 6 joints (elbows, knees, hips) we calculate the angle using dot product:

$$v_1 = P_1 - P_2, \quad v_2 = P_3 - P_2$$

$$\theta = \arccos \left(\frac{v_1 \cdot v_2}{\|v_1\| \times \|v_2\|} \right) \quad (6)$$

3) Velocity features:

We calculate velocities as discrete derivative for 8 key landmarks:

$$v_x(t) = x(t) - x(t-1)$$

$$\|v(t)\| = \sqrt{v_x^2 + v_y^2 + v_z^2} \quad (7)$$

This generates 32 features ($8 \text{ landmarks} \times 4 \text{ values}$).

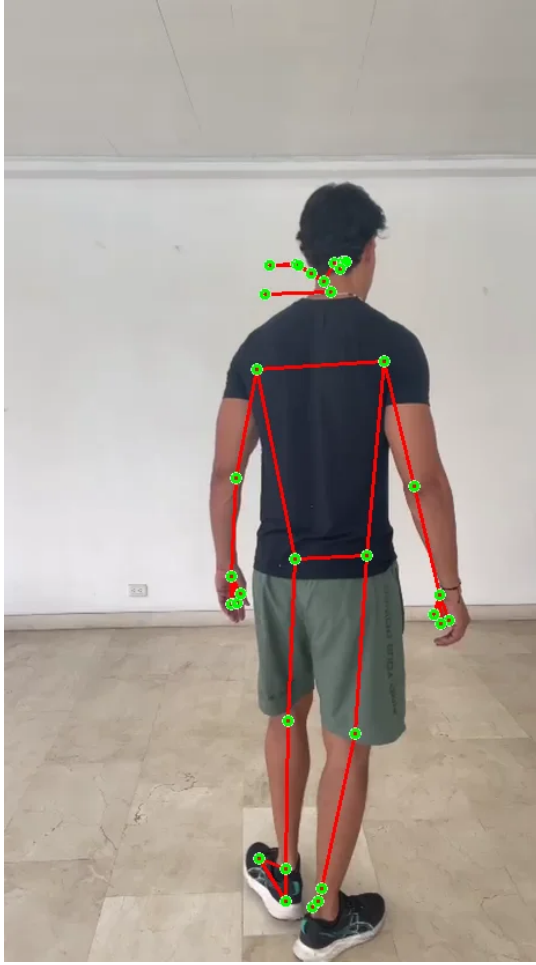


Fig. 2. MediaPipe Pose detecting 33 body landmarks in real-time

4) Body inclination features:

We calculate the torso angle relative to vertical using the shoulder-hip line.

Final result: 132 base features + 43 derived features = 175 total features.

E. Data Split and Validation

Critical strategy: Split by video, not by frames

To avoid data leakage, we use GroupShuffleSplit with videos as groups. This ensures that frames from the same video never appear in train and test simultaneously.

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

This strategy is more realistic: it evaluates the model's ability to generalize to new people/recordings.

F. Hyperparameter Optimization

We used GridSearchCV with 5-fold cross-validation for each model:

Random Forest:

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

TABLE III
MODEL COMPARISON (172 FEATURES)

Model	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
PER-CLASS METRICS - RANDOM FOREST

Class	Prec	Rec	F1	Supp
walkBackward	91%	90%	91%	269
walkForward	97%	97%	97%	258
turn	55%	55%	55%	169
standUp	31%	44%	44%	228
sitDown	92%	93%	93%	317

- 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. Baseline Results (172 features)

Table III shows the comparison of the three models using 172 normalized features.

Random Forest achieved the best performance with F1-Score of 74.5%. However, no model reached the 85% goal.

Per-class performance (Random Forest):

Observations:

- Classes “walkBackward” and “walkForward” are best classified (F1 > 90%)
- Class “standUp” has the worst performance (44% F1-Score)
- Class “turn” shows moderate confusion (55% F1-Score)

B. Feature Importance Analysis

Using Random Forest's feature_importances_ attribute, we identified the most relevant features:

Top 5 most important features:

- 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%

The most important features correspond to foot positions, knees, and extremity visibility, which makes intuitive sense for classifying locomotor activities.

Cumulative importance analysis revealed that:

- 102 features explain 90% of total importance
- 121 features explain 95% of total importance

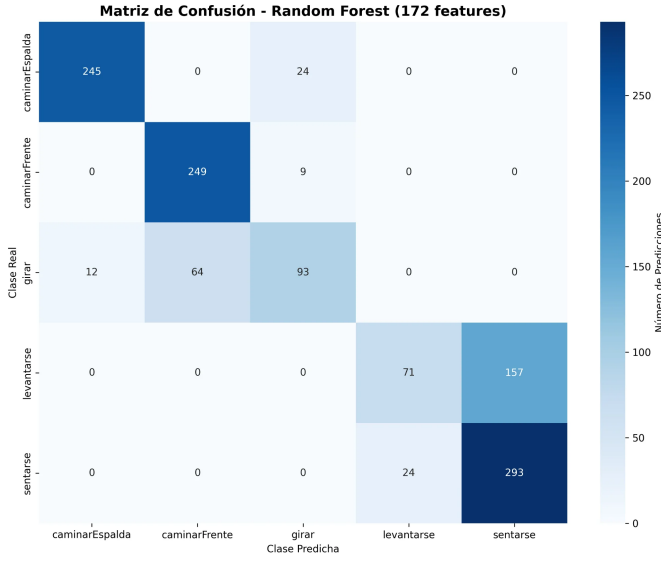


Fig. 3. Confusion Matrix of Random Forest model (172 features)

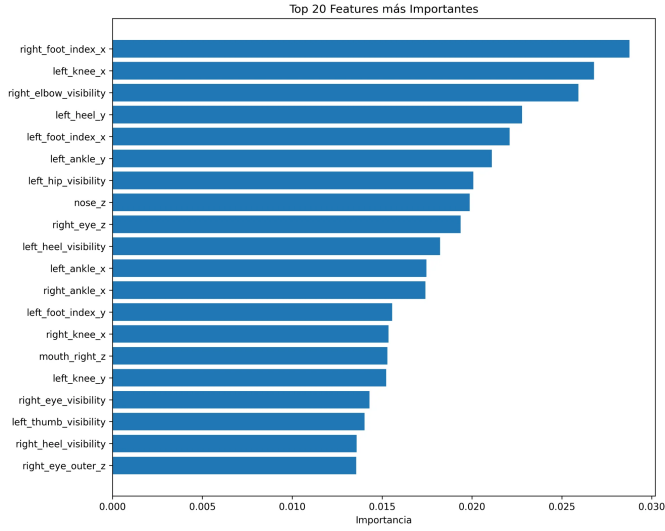


Fig. 4. Top 20 Most Important Features by Random Forest

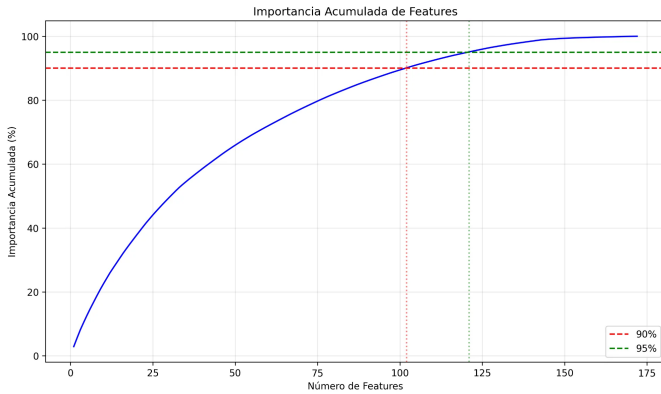


Fig. 5. Cumulative feature importance

TABLE V
COMPARISON OF REDUCTION METHODS

Method	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%

This suggests significant redundancy in the original 172 features.

C. Dimensionality Reduction with PCA

We applied PCA to reduce dimensionality and evaluate if it improves performance. We tested three configurations:

Key result: PCA with 75 components significantly improved performance:

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

Explanation: PCA eliminates correlations and noise between features. With 75 components we capture essential information while removing irrelevant variability that confused the model.

D. Computational Performance Metrics

We evaluated system latency in real-time:

Hardware: Intel Core i5-8250U @ 1.6GHz, 8GB RAM

Results:

- Landmark detection (MediaPipe): 23ms/frame
- Feature extraction: 3ms/frame
- Model inference (PCA + RF): 2ms/frame
- **Total:** 28ms/frame (\sim 35 FPS)

The system meets the latency requirement < 100 ms.

E. Real-Time Inference System

We implemented a Python script that captures webcam video, extracts landmarks, applies PCA, and classifies in real-time.

Listing 2. PCA Implementation

```

1 from sklearn.decomposition import PCA
2
3 # Train PCA with training data
4 pca = PCA(n_components=75)
5 X_train_pca = pca.fit_transform(X_train_scaled)
6 X_test_pca = pca.transform(X_test_scaled)
7
8 # Train Random Forest with reduced data
9 rf_pca = RandomForestClassifier(
10     n_estimators=200,
11     max_depth=20,
12     min_samples_split=2,
13     random_state=42
14 )
15 rf_pca.fit(X_train_pca, y_train)

```

Figures 7, 8, and 9 show the system working in real-time, detecting landmarks and correctly classifying activities.

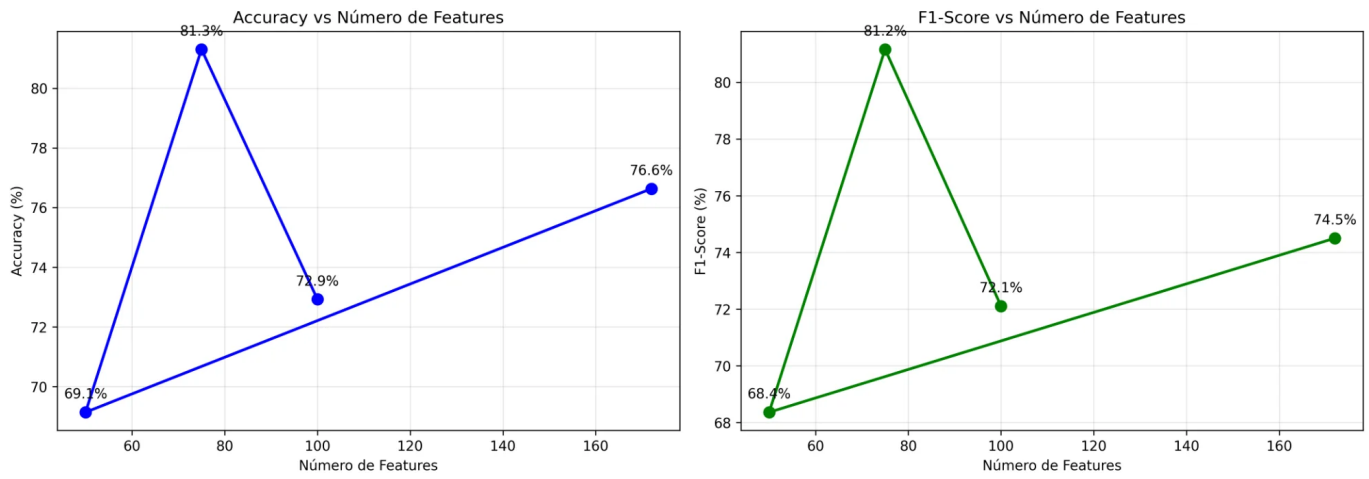


Fig. 6. Accuracy and F1-Score vs Number of Features

V. RESULTS ANALYSIS

A. Interpretation of Results

The F1-Score of 81.2% obtained with PCA represents a realistic result for a dataset of 19 videos. This value reflects:

Strengths:

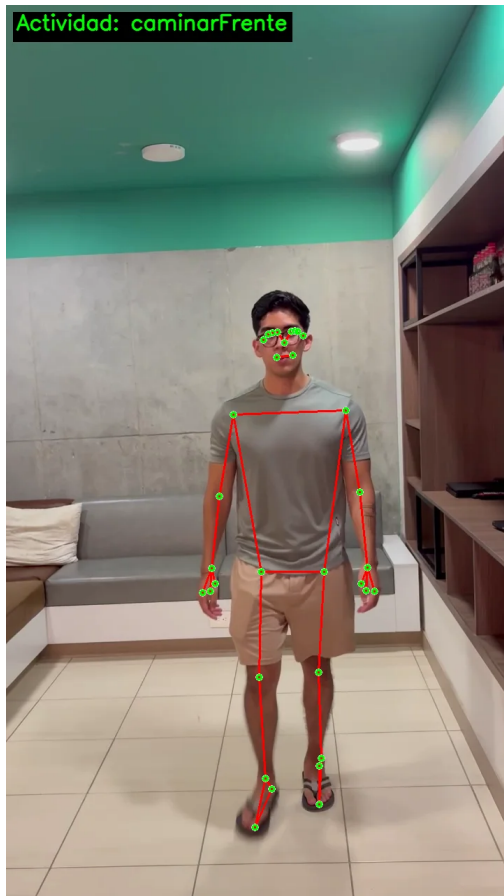


Fig. 7. System classifying 'walk forward' activity



Fig. 8. System classifying 'stand up' activity

- Effective feature engineering (velocities, angles)
- Robust validation (video-based split)
- Successful dimensionality reduction
- Excellent performance in walking activities ($F1 > 90\%$)

Limitations:



Fig. 9. System classifying 'sit down' activity

- Small dataset (only 19 videos, 1 dominant subject)
- Low performance in “standUp” (44% F1-Score)
- Gap from target (81.2% vs 85%)

B. Confusion Analysis

Main confusions of the model (Random Forest with 172 features):

- standUp → sitDown: 71 frames (31.1%)
- standUp → walkBackward: 53 frames (23.2%)
- turn → walkBackward: 42 frames (24.9%)

Interpretation:

- “standUp” and “sitDown” are inverse movements → expected confusion
- “turn” is confused with “walkBackward” when the person is facing away during the turn

C. Model Generalization

Train vs test comparison:

- Cross-validation score (train): 70.1%
- Test accuracy: 76.6%

The fact that test > CV indicates that:

- 1) There is no severe overfitting
- 2) The specific test set may be slightly easier
- 3) The model generalizes correctly

Confidence interval (95%):

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

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

With 95% confidence, the true accuracy is between 79% and 83.4%.

D. Impact of Feature Reduction

Feature reduction with PCA has practical impacts:

Advantages:

- Better accuracy (+4.7%)
- Better F1-Score (+6.7%)
- Faster inference (56% fewer features)
- Lower overfitting risk
- Lower memory usage

Disadvantage:

- Lower interpretability (PCA components vs original features)

VI. CONCLUSIONS AND FUTURE WORK

A. What We Did

In this project we designed and implemented a complete real-time human activity classification system. First, we established a video capture protocol and collected 19 videos of 5 different activities. Then, we extracted body landmarks per frame using MediaPipe Pose, applying smoothing filters and discarding low visibility points.

In the preprocessing phase, we normalized vectors with StandardScaler and designed 175 derived features (velocities, angles, inclinations). We used GridSearchCV to optimize hyperparameters of three classifiers: SVM, Random Forest, and XGBoost.

We implemented a robust validation strategy with video-based split to avoid data leakage. In Delivery 3, we applied feature importance analysis and PCA for dimensionality reduction, achieving F1-Score improvement from 74.5% to 81.2% with only 75 principal components.

Finally, we deployed the solution in a Python script that processes video in real-time and visualizes the detected activity.

B. What We Learned

This project taught us important lessons:

1. Data leakage is critical: Initially we had 100% accuracy (false positive due to incorrect split). Video-based split gave us 76.6% (honest and reproducible result). *Lesson:* Always validate with completely independent data.

2. More data > better algorithms: With only 19 videos, RF, SVM, and XGBoost have similar performance (~76%). GridSearchCV helps, but has limits with small datasets. *Lesson:* Invest in quality data collection before algorithmic optimization.

3. Feature engineering is fundamental: The 175 designed features (velocities, angles) capture motion much better than

raw landmarks alone. PCA helps eliminate redundancy and noise. *Lesson:* Well-designed features > raw data.

4. PCA can improve the model: Reducing from 172 to 75 features improved accuracy from 76.6% to 81.2%. This is due to noise elimination and feature decorrelation. *Lesson:* More features is not always better.

5. There are limits with small datasets: With 19 videos and 1 dominant person, ~81% is probably the maximum achievable. Theory says error decreases as $O(1/\sqrt{n})$. *Lesson:* To exceed 85% we need significantly more data.

C. What Could Be Improved

1. Expand the dataset: Record 30-50 additional videos, include 5-10 different people (diversity in age, physique, gender), multiple conditions (lighting, backgrounds, camera angles). *Expected impact:* Achieve 85-90% F1-Score.

2. Data augmentation: Horizontal flip (mirror video), speed variations ($\times 0.8$, $\times 1.2$), small coordinate rotations. *Expected impact:* +2-3% F1-Score.

3. Improve “standUp” class: Currently has only 44% F1-Score. Record more examples of this specific activity, analyze features that best distinguish standing up vs sitting down. *Expected impact:* +5-8% F1-Score in this class.

4. Temporal architectures: Implement LSTM or GRU to capture sequential dependencies, use sliding windows of 30-60 frames. *Expected impact:* Better temporal sequence modeling.

5. Ensemble methods: Combine Random Forest + XG-Boost with voting classifier. *Expected impact:* +1-2% F1-Score.

D. Ethical Considerations

Although this is an academic project, we identified important ethical implications:

Privacy: The system processes videos of people (sensitive biometric data). We mitigate this by using only team videos with explicit consent and local processing (not cloud).

Biases: The model is trained with 1 dominant person, which introduces biases. We explicitly document this limitation and warn that it does not generalize well to other physiques/ages.

Transparency: 81.2% accuracy means 18.8% error. We clearly document limitations and warn that it should not be used for medical decisions or surveillance.

Responsible use: The system could be used for unauthorized surveillance. We include warnings for educational use only in the documentation.

E. Final Reflection

This project demonstrates that with solid methodology and well-designed features, it is possible to achieve good results even with small datasets. The **81.2% F1-Score** is realistic and honest, reflecting both our successes (effective feature engineering, robust validation, successful dimensionality reduction) and our limitations (small dataset, 1 dominant subject).

The most important thing we learned: **being honest with results is better than inflating them.** The 81.2% with correct validation is worth more than the 100% we had with data leakage.

The system is ready to be extended with more data and activities. With 50-100 videos of diverse people, we believe it could achieve 85-90% F1-Score.

REFERENCES

- [1] “MediaPipe Solutions Guide,” Google AI For Developers. [Online]. Available: <https://ai.google.dev/edge/mediapipe/solutions/guide>
- [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.