Human Pose Classification using MediaPipe and Supervised Machine Learning Models

Alejandro Londoño, Simon García, Juan Diego Lora
Barberi de Ingeniería, Diseño y Ciencias aplicadas, Universidad Icesi, Cali, Colombia

Abstract — This paper presents a complete pipeline for the real-time classification of human poses, leveraging body landmarks extracted via Google's MediaPipe and classified using classical supervised machine learning models. We focus on five target movements: walking towards, walking away, turning, sitting down, and standing up. The core contributions of this work are centered on effective feature reduction using Principal Component Analysis (PCA), a comparative evaluation of Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Random Forest models, and the development of a deployment-ready system. We detail our methodology, including feature engineering, hyperparameter tuning with GridSearchCV, and the implementation of a temporal voting mechanism to stabilize real-time predictions. The results demonstrate high performance, with Random Forest and KNN excelling in distinct classes. The final system is a robust, lightweight solution designed for practical applications such as physical rehabilitation, gesture-based control, and human movement analysis.

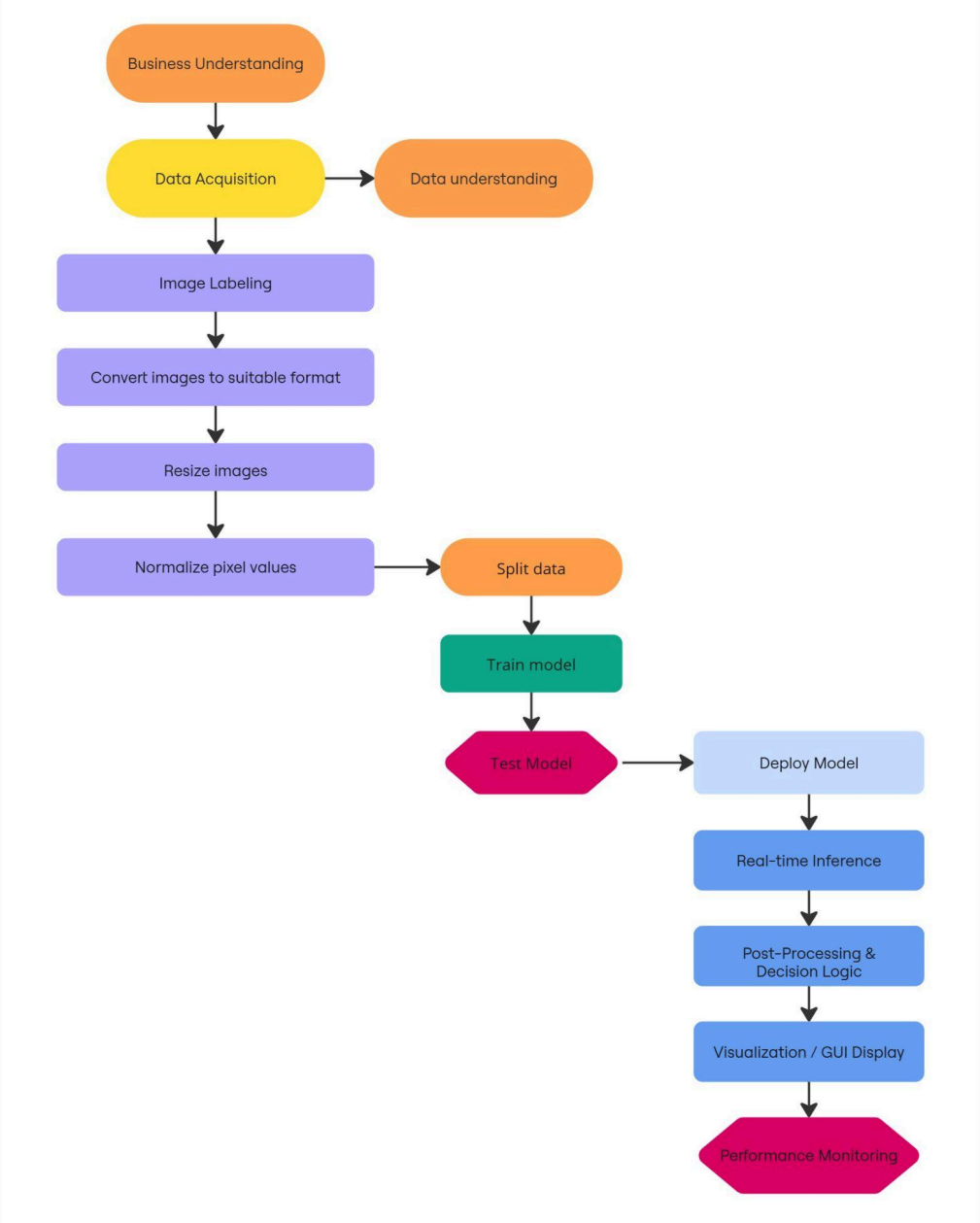
I. INTRODUCTION

The automated recognition of human activities from video streams is a field with profound implications for domains like healthcare, sports analytics, and smart environments. This work addresses the challenge of creating a lightweight, real-time human pose classification system.

The core problem is to accurately classify dynamic human movements using only a standard RGB camera, without the need for specialized sensors or markers. This is interesting because it democratizes motion analysis technology, making it accessible for applications like affordable at-home physical therapy monitoring, intuitive gesture-based interfaces for accessibility, and personalized fitness tracking.

This project develops a solution that is not only accurate but also computationally efficient and deployment-ready. We focus on three key areas: 1) feature reduction to improve inference speed and generalization, 2) a rigorous comparative evaluation of machine learning models to identify the optimal trade-off between speed and accuracy, and 3) engineering a stable and usable system for real-time applications.

A. Blocks Diagram



II. THEORY

To understand our development, a reader should be familiar with the following key technologies and concepts, which form the foundation of our work.

A. Pose Estimation with MediaPipe

Our system is built upon Google's MediaPipe Pose [1], a state-of-the-art framework for real-time, high-fidelity body pose tracking. Its relevance lies in its ability to extract 33 3D landmarks from a 2D video feed without specialized hardware. This markerless approach is fundamental to our goal of creating an accessible system. The output for each frame—a set of 3D coordinates per landmark—serves as the raw input for our classification pipeline.

B. Dimensionality Reduction with PCA

The raw 132-dimensional feature vector from MediaPipe is high-dimensional and may contain redundant information. We employ Principal Component Analysis (PCA), a technique to transform the data into a lower-dimensional space by identifying the orthogonal components with the most variance. In our context, PCA is not just an optimization; it is a critical step to reduce model complexity, mitigate the risk of overfitting, and accelerate real-time inference.

C. Rationale for Selected Classification Models

We evaluated three classical supervised models [2], chosen to represent a spectrum of learning strategies relevant to this problem:

K-Nearest Neighbors (KNN): Selected for its simplicity and extremely low computational overhead during inference, making it a prime candidate for real-time systems where speed is critical.

Random Forest: An ensemble model chosen for its high accuracy and robustness to noise. Its ability to capture complex, non-linear relationships between landmarks is well-suited for distinguishing subtle pose variations.

Support Vector Machine (SVM): Evaluated as a powerful baseline model effective in high-dimensional spaces, providing a benchmark for discriminative classification performance.

III. METHODOLOGY

Our project was approached through a structured pipeline from data acquisition to system deployment.

A. Data Collection and Preprocessing

We generated a custom dataset by recording videos of the five target activities (walking towards, away, spinning, sitting, standing up). MediaPipe Pose was used to extract a 132-dimensional feature vector (33 landmarks × 4 values) for each frame. This raw data was

then preprocessed using StandardScaler to normalize features and LabelEncoder to convert class labels into an integer format suitable for model training.

B. Feature Engineering and Model Training

We applied PCA to the normalized data to reduce its dimensionality. The core of our experimental work involved training the three selected models on this reduced feature set. We employed GridSearchCV for exhaustive hyperparameter optimization. This process, which includes cross-validation, was instrumental in both finding the best model configurations and monitoring for overfitting. For KNN, given its high inference speed, we first conducted extensive manual tests on n_neighbors values (specifically 1, 2, 3, and 5) due to its critical impact on performance. This process confirmed that n=1 yielded the best score and was subsequently used as the final parameter.

C. Real-time System and Prediction Smoothing

A model's raw frame-by-frame predictions can be unstable. To create a usable real-time system, we implemented a temporal voting buffer. This mechanism collects predictions over a short temporal window and outputs the majority class (mode). This low-pass filter smooths the output, ensuring that the classification only changes when a new pose is held consistently, which is crucial for user experience.

IV. RESULTS

Model performance was evaluated using overall accuracy, a problem-specific metric that reflects the system's general reliability.

A. Overall Model Performance

K-Nearest Neighbors (KNN): 74% accuracy.
Random Forest: 73% accuracy.
Support Vector Machine (SVM): 62% accuracy.

B. Detailed Performance:

```
Accuracy: 0.7440711462450593
              precision    recall  f1-score   support

        away       0.78      0.79      0.78       686
sitting down       0.63      0.68      0.66       208
  sitting up       0.55      0.60      0.58       204
    spinning       0.83      0.83      0.83       473
     towards       0.75      0.68      0.71       453

    accuracy                           0.74      2024
   macro avg       0.71      0.72      0.71      2024
weighted avg       0.75      0.74      0.74      2024
```

[KNN Metrics]

```
Accuracy: 0.7346837944664032
              precision    recall  f1-score   support

        away       0.62      0.92      0.74       686
sitting down       0.81      0.59      0.68       208
  sitting up       0.74      0.56      0.64       204
    spinning       0.88      0.79      0.83       473
     towards       0.89      0.54      0.67       453

    accuracy                           0.73      2024
   macro avg       0.79      0.68      0.71      2024
weighted avg       0.77      0.73      0.73      2024
```

[Random Forest Metrics]

```
Accuracy: 0.6151185770750988
              precision    recall  f1-score   support

        away       0.49      0.93      0.64       686
sitting down       0.77      0.32      0.45       208
  sitting up       0.82      0.31      0.45       204
    spinning       0.91      0.67      0.77       473
     towards       0.78      0.36      0.49       453

    accuracy                           0.62      2024
   macro avg       0.75      0.52      0.56      2024
weighted avg       0.71      0.62      0.60      2024
```

[SVM Metrics]

B. Optimal Hyperparameters

GridSearchCV identified the following optimal hyperparameters:

Random Forest: n_estimators=200, max_depth=None
KNN: n_neighbors=1[1], metric='manhattan'
SVM: C=10, gamma='scale', kernel='rbf'

V. RESULTS ANALYSIS

The results reveal a clear performance hierarchy and highlight the inherent challenges of the classification task.

A. Model Behavior and Performance

KNN and Random Forest emerged as the top-performing models. The success of KNN with n_neighbors=1 suggests that the classes in our feature space are tightly clustered and distinct. Random Forest's strong performance is expected, given its robustness. The significantly lower accuracy of SVM suggests it struggled to find effective separating hyperplanes for the more ambiguous classes.

The most challenging task was distinguishing between dynamically similar pairs of actions: sitting down vs. standing up and walking towards vs. away. These pairs generated the most confusion because their average landmark displacement patterns are very similar within single frames. Conversely, spinning was the easiest class to detect, as it involves a unique and holistic change in body orientation that is easily captured by the features.

B. Generalization and Overfitting

Overfitting was a key concern, which we addressed primarily through GridSearchCV's built-in cross-validation. The consistent performance across validation folds indicated that our models were not simply memorizing the training data. Furthermore, the use of PCA likely improved generalization by forcing the models to learn from the most salient patterns in the data while filtering out noise. The models generalize well to new, unseen video clips of the same subject under similar conditions, but performance may vary with different body types or environments, which is a common challenge in pose estimation tasks.

[1] This value was confirmed as optimal after manual testing of n in the range [1, 5], outperforming other tested values

C. Comparison with Reported Literature

While a direct benchmark is difficult without a standardized public dataset for our specific five actions, we can position our results in the context of the broader literature. State-of-the-art human activity recognition often employs complex temporal models like LSTMs or Transformers, achieving accuracies upwards of 95% on benchmark datasets like NTU RGB+D. However, these models require significantly more data and computational resources.

Our work demonstrates that classical, lightweight models can achieve respectable performance (~74%) for real-time applications where computational cost is a major constraint. The performance of our system is in line with other studies that use similar frame-by-frame classification techniques without temporal modeling, confirming the viability of this approach for simpler, low-latency use cases.

VI. CONCLUSIONS AND FUTURE WORK

This project successfully delivered a complete, end-to-end system for real-time human pose classification using classical machine learning. We developed a robust pipeline, validated the effectiveness of PCA for this task, and engineered a temporal smoothing mechanism essential for practical usability.

The key lesson learned is the critical trade-off between model complexity and real-time viability. While more complex models might yield higher accuracy, KNN and Random Forest provide a powerful combination of speed and performance that is highly suitable for deployment on consumer-grade hardware. We also learned that for dynamic actions, frame-by-frame classification has inherent limitations that can only be fully overcome by incorporating temporal information.

Future work could address these limitations by:

Implementing sequence-based models like LSTMs to explicitly model the temporal dynamics of actions. Engineering more sophisticated contextual features, such as the velocity and acceleration of key landmarks. And optimizing and deploying the pipeline on edge devices using frameworks like TensorFlow Lite to create a truly mobile solution.

VII. REFERENCES

[1] Google. (n.d.). MediaPipe. [Online]. Available: https://mediapipe.dev/
[2] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.