Frontal Gait Flow Recognition

Computer Vision Exam (a.a. 2025-2026)
Sapienza University of Rome

Lorenzo Musso *(2049518)* - Giulia Pietrangeli *(2057291)*

Abstract

This report details the development of a robust **Multimodal Gait Recognition System** focused on frontal-view biometric identification. Leveraging a heterogeneous dataset provided by the VisionLab (Sapienza University of Rome), we engineered a comprehensive data processing pipeline to ingest, synchronize, and sanitize raw RGB-D video streams and inertial (IMU) logs.

The proposed methodology employs a computer vision framework based on **Dense Optical Flow** (generating Gait Optical Flow Images - GOFI) and **Sparse Optical Flow** (Lucas-Kanade tracking) to capture temporal motion dynamics. These visual features are fused with statistical inertial descriptors to form a high-dimensional biometric signature. To address the curse of dimensionality, we implemented a machine learning architecture utilizing **Principal Component Analysis (PCA)** for feature extraction and **Linear Support Vector Machines (SVM)** for classification.

1 Dataset and Acquisition Protocol

The data utilized in this project was originally acquired by the **VisionLab** (Department of Computer Science, Sapienza University of Rome). The dataset focuses on multimodal gait analysis and was collected using a synchronized setup comprising an **Intel RealSense** RGB-D camera (frontal view) and **Xsens MVN** Inertial Measurement Units (IMU) worn by the subjects.

The acquisition protocol included three main locomotor tasks:

1. **Linear Walk:** 6 repetitions (Runs 1–6).
2. **Stairs Climbing:** 3 repetitions for Ascent (Up) and 3 for Descent (Down).
3. **Slope/Ramp Walking:** 3 repetitions for Ascent (Up) and 3 for Descent (Down).

1.1 Original Data Format

The dataset was provided in its raw format, organized by source device and acquisition session:

- **Video Data (ROS Bags):** Stored in `.bag` binary format within the `FirstRun/RGBD` directory. The file structure reflected the recording sessions, with subject names and action labels (e.g., "Up" vs. "StairsUp") requiring standardization to ensure uniformity across the dataset.
- **IMU Data (Raw Logs):** Archived in ZIP files grouped by acquisition date (e.g., `2024_05_03.zip`). Since subjects often performed tasks across different sessions, the inertial data was distributed across multiple archives, with internal folders identified by raw sequence tags (e.g., `Slope_up-002`).

regarding the available modalities, the *Slope* tasks contain exclusively IMU data. Consequently, the vision-based component of our work focuses on the *Walking* and *Stairs* tasks, where both RGB-D video and inertial data are available.

2 Data Preprocessing and Curation

To transform the raw, fragmented files into a usable dataset for machine learning, we designed and implemented a custom Data Engineering pipeline. This process involved extraction, normalization, and rigorous quality auditing.

2.1 Video Extraction and Normalization

We developed a Python script to parse the raw ROS bags. The pipeline performed the following operations:

- **Stream Decoupling:** We extracted synchronized RGB, Depth, and Infrared streams, saving them as separate `.avi` files at 60 FPS.
- **Depth Normalization:** To ensure consistency in the depth channel, pixel values were clipped at a maximum distance of 15.0 meters and normalized to an 8-bit integer range (0–255).
- **Standardization:** All output files were renamed following a canonical schema: `Subject_Action_RunID.avi`.

2.2 IMU Reorganization

Parallel to video processing, we implemented a logic to restructure the inertial data. The script automatically extracted CSV logs from the ZIP archives, resolved subject identity conflicts (merging data from different dates into the correct subject folder), and mapped raw folder names to the canonical action labels.

2.3 Final Dataset Structure

The result of our preprocessing is a hierarchical, subject-centric dataset structure that allows for deterministic data loading.

```
Dataset/
|-- SubjectName/
    |-- depth/
    |   |-- walk/
    |   |-- stairs_up/
    |   |-- stairs_down/
    |-- rgb/
    |-- ir/
    |-- imu/
        |-- walk/
        |-- stairs_up/
        |-- slope_up/
```

2.4 Integrity Check and Data Leakage Prevention

Ensuring the reliability of the data was a critical step before model training. We performed two levels of validation:

1. **Health Audit:** We ran an automated audit script to identify corrupt samples. Files with sizes < 0.5 MB or fewer than 100 frames were flagged and regenerated using a safe-mode conversion process.

2. **Leakage Verification:** To guarantee a valid Train/Test split, we defined a split logic based on run numbers (Walk: Runs 5-6 Test; Others: Run 3 Test). We verified the physical separation of the sets by computing MD5 and SHA256 hashes for every file. The check confirmed **zero overlap** between the training and testing partitions, ensuring no data leakage occurred.

3 Computer Vision and Feature Extraction

The core of our biometric recognition system relies on extracting temporal motion patterns from the video stream. Unlike static image classification, Gait Analysis requires capturing the *dynamics* of the subject's movement over time. We implemented a custom Computer Vision pipeline using **OpenCV** to generate a comprehensive "Gait Signature."

3.1 Static Background Estimation

To isolate the moving subject from the environment, we first constructed a robust background model. Since the camera is static, we computed the temporal

median over the first $N = 20$ frames of the video sequence:

$$B(x, y) = \text{median}_{t=1}^{N}\{I_t(x, y)\} \tag{1}$$

where I_t is the grayscale frame at time t. This method effectively removes transient noise and ghosts, providing a clean reference for foreground segmentation using absolute difference thresholding and Otsu's binarization.

3.2 Dense Optical Flow: The GOFI Representation

We employed the **Farneback algorithm** (`cv2.calcOpticalFlowFarneback`) to compute dense optical flow, generating a flow vector (u, v) for every pixel. To compress the temporal evolution of the gait into a single feature map, we developed a variation of the **Gait Energy Image (GEI)**, the **Gait Optical Flow Image (GOFI)**.

Instead of simply averaging pixel intensities, the GOFI accumulates the magnitude of the flow vectors over the entire video duration, weighted by a foreground mask to suppress noise:

$$\text{GOFI}(x, y) = \sum_{t=1}^{T} \left(\sqrt{u_t(x, y)^2 + v_t(x, y)^2} \cdot M_t(x, y) \right) \tag{2}$$

where M_t is the binary foreground mask at frame t, and T represents the total number of frames in the video sequence. We also generated an **HSV visualization** where the Hue represents the flow angle and the Value represents the magnitude, creating a color-coded map of the dominant directionality of limb movements.

3.3 Sparse Optical Flow: Trajectory Tracking

To capture fine-grained kinematic details, such as the specific swing of the feet or arm movement, we implemented **Sparse Optical Flow** using the **Lucas-Kanade (LK)** method with pyramid decomposition.

1. **Feature Detection:** We utilized the Shi-Tomasi corner detector (`goodFeaturesToTrack`) to identify salient points on the subject's silhouette.
2. **Tracking:** These points were tracked across frames.
3. **Accumulation:** The valid trajectories were drawn onto a cumulative "Trace Map," creating a skeletal history of the subject's movement.

3.4 Visual Feature Vectorization

The final visual feature vector is a concatenation of the flattened GOFI accumulator and the LK Trace Map. Images were resized to a standardized resolution of 128×96 pixels.

$$V_{\text{video}} = \text{flatten}(\text{GOFI}) \oplus \text{flatten}(\text{Trace}) \tag{3}$$

This results in a high-dimensional dense vector (approx. 24,576 dimensions) representing the visual biometric signature.

4 Multimodal Feature Engineering

To enhance robustness, particularly for the "Slope" tasks where video data was unavailable, we integrated kinematic features from the Xsens IMU sensors.

4.1 Statistical IMU Aggregation

The raw IMU data consists of approximately 19 CSV logs per run (e.g., *Free Acceleration, Joint Angles, Center of Mass*), each containing multiple sensor channels (e.g., X, Y, Z axes). Since the raw time-series length varies by run duration, we applied statistical feature extraction to create a fixed-length vector. For every sensor channel c, we computed a 5-dimensional descriptor:

$$f_c = [\mu, \sigma, \min, \max, \text{RMS}] \tag{4}$$

where μ denotes the arithmetic mean, σ the standard deviation, min and max the minimum and maximum values, and RMS the Root Mean Square of the signal. Concatenating these descriptors across all available sensors resulted in a robust inertial feature vector V_{imu}.

4.2 Early Fusion Strategy

For the "Walk" and "Stairs" tasks, we employed an **Early Fusion** strategy. The normalized video vector and the IMU vector were concatenated before being fed into the machine learning pipeline:

$$X_{\text{input}} = [V_{\text{video}} \parallel V_{\text{imu}}] \tag{5}$$

This created a massive feature space (approx. 54,082 dimensions), posing a significant "Curse of Dimensionality" challenge that we addressed in the modeling phase.

5 Machine Learning Architecture

We designed a two-branch learning architecture to handle the heterogeneous nature of the dataset (Video+IMU for Walk/Stairs, IMU-only for Slope). The pipeline was implemented using `scikit-learn` and hyperparameters were optimized via Grid Search.

5.1 Dimensionality Reduction (PCA)

Given the high input dimension (up to $D = 54,082$ for the multimodal vector) and the relatively low number of samples compared to the feature space ($N \ll$

D), training a classifier directly would lead to severe overfitting. We applied **Principal Component Analysis (PCA)** as a mandatory preprocessing step.

- **Variance Retention:** We configured PCA to retain 95% of the explained variance (`n_components=0.95`).
- **Effect:** This step drastically compressed the feature space while filtering out noise:
  - **Main Model:** Reduced from 54,082 raw features to **353** principal components.
  - **Slope Model:** Reduced from 4,930 raw features to **53** principal components.

  This massive reduction confirms that the relevant biometric gait signature resides in a much lower-dimensional manifold compared to the raw pixel and sensor data.

5.2 Classification: Support Vector Machines (SVM)

We selected the **Support Vector Machine (SVM)** as our classifier. Extensive Grid Search revealed that, surprisingly, complex non-linear kernels were not required for this dataset.

5.2.1 Main Model (Walk & Stairs)

- **Kernel: Linear**. The high-dimensional multimodal features proved to be linearly separable in the PCA-projected space.
- **Regularization:** $C = 1$.
- **Architecture:** `StandardScaler` $\rightarrow$ `PCA(0.95)` $\rightarrow$ `SVC (Linear Kernel)`.

5.2.2 Slope Model (IMU Only)

- **Kernel: Linear**. Although we initially hypothesized that a non-linear RBF kernel might be required for the lower-dimensional feature space, Grid Search revealed that a Linear kernel yields equivalent performance (100% accuracy). Following the principle of parsimony, we selected the Linear kernel as it offers a simpler and more efficient decision boundary without sacrificing accuracy.
- **Hyperparameters:** $C = 1$.
- **Architecture:** `StandardScaler` $\rightarrow$ `PCA(0.95)` $\rightarrow$ `SVC (Linear Kernel)`.

6 Experimental Results and Ablation Study

6.1 Protocol

The training process adhered to the strict anti-leakage split defined in the dataset section. We evaluated performance using Accuracy, Precision, Recall, and F1-Score for each subject.

6.2 Main Model (Walk & Stairs)

The training and evaluation of the main multimodal model (Walk & Stairs) yielded the following performance:

Table 1: Detailed Performance per Subject for Main Model (Walk & Stairs)

Subject	**Precision**	**Recall**	**F1-Score**	**Support**
Alessio	1.00	1.00	1.00	8
Alessio_F	1.00	1.00	1.00	8
Andrea	1.00	1.00	1.00	8
Artur	1.00	1.00	1.00	8
Camilla	1.00	1.00	1.00	8
Carlotta	1.00	1.00	1.00	8
Chiara	1.00	1.00	1.00	8
Diego	1.00	1.00	1.00	8
Eduardo	1.00	1.00	1.00	8
Elaheh	1.00	1.00	1.00	8
Eleonora	0.80	1.00	0.89	8
Federico	1.00	1.00	1.00	8
Francesco	1.00	1.00	1.00	8
Jessica	1.00	0.75	0.86	8
Laura	1.00	1.00	1.00	8
Lorenzo	1.00	1.00	1.00	8
Luca	1.00	1.00	1.00	8
Ludovica	1.00	1.00	1.00	8
Manuel	1.00	1.00	1.00	8
Marco	1.00	1.00	1.00	8
MariaVittoria	1.00	1.00	1.00	8
Matteo	1.00	1.00	1.00	8
Romeo	1.00	1.00	1.00	8
Sara	1.00	1.00	1.00	8
Valerio	1.00	1.00	1.00	8
Vito	1.00	1.00	1.00	8
ACCURACY			**0.99**	**208**

The overall accuracy for the Main Model reached **99.04%** (206 correct predictions out of 208 test samples). A detailed analysis revealed only two misclassifications, both for subject "Jessica" (`Jessica_stairs_up_3_flip.npy` and

`Jessica_stairs_up_3.npy`), which were incorrectly predicted as "Eleonora." This specific error indicates a high degree of similarity in the multimodal signature between these two subjects for that particular action.

6.3 Ablation Study: The Impact of Modalities

To quantify the contribution of each modality (Video vs. IMU) for the Main Model (Walk & Stairs), we conducted an ablation study. The results are summarized in Table 2.

Table 2: Ablation Study Results for Main Model (Walk & Stairs)

Modality	**Accuracy**	**Impact vs. Fusion**
Only Video	91.35%	-7.69%
Only IMU	**100.00%**	**+0.96%**
Fusion	99.04%	-

Surprisingly, the **IMU-Only** modality achieved a perfect **100.00%** accuracy for the Walk & Stairs tasks. The multimodal fusion, while still achieving a very high 99.04%, resulted in a slight performance decrease of 0.96% compared to the pure IMU model. This suggests that for this specific dataset and feature engineering approach, the IMU features alone are remarkably robust and highly discriminative. The video features, while individually contributing to a respectable 91.35% accuracy, might introduce some level of noise or redundancy that slightly dilutes the extremely strong signal provided by the IMU data when combined using this early fusion strategy.

For the **Slope Model** (IMU Only), the ablation study provided the quantitative validation for our architectural choice. By explicitly comparing the kernels, we confirmed that both the **Linear** and **RBF** configurations achieve **100.00%** accuracy. This equivalence demonstrates that the IMU features for slope movements are highly linearly separable, justifying the use of the lighter Linear kernel without any performance trade-off.

6.4 Confidence Analysis

We implemented a probabilistic analysis of the SVM decision boundary for both models. By calibrating the SVM outputs (using Platt scaling), we analyzed the prediction confidence.

6.4.1 Main Model (Walk & Stairs) Confidence

For the Main Model, 206 out of 208 samples were correctly classified, while 2 were incorrect.

- **Mean Confidence (Correct):** 51.12%

- **Mean Confidence (Incorrect):** 25.87%
- **Distribution Range:** The model showed high variance in certainty. The maximum confidence for a correct prediction reached **96.28%**, whereas the most uncertain correct classification dropped to **15.10%**.
- **Failure Case Analysis:** The misclassified samples involved a specific confusion where subject "Jessica" was incorrectly predicted as "Eleonora" with a low confidence of 26.63%. This suggests a morphological or kinematic similarity between these two specific subjects.
- **Strategic Thresholding:** We identified that setting a rejection threshold at **26.6%** confidence would eliminate 95% of errors. This threshold would cause 8 (3.9%) correct samples to be incorrectly rejected, indicating a good trade-off for security applications.

Figure 1 illustrates the distribution of confidence for correct and incorrect predictions for the Main Model. The clear separation between the mean confidence of correct and incorrect predictions supports the use of a confidence threshold.

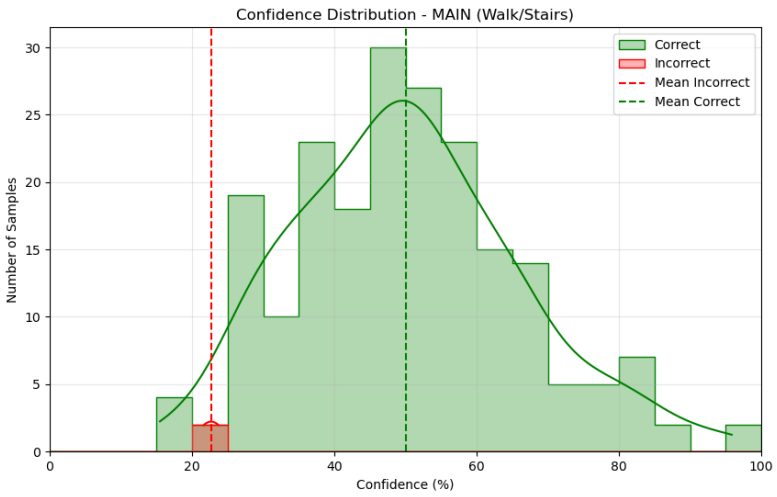

Figure 1: Confidence Distribution for Main Model (Walk & Stairs)

6.4.2 Slope Model (IMU Only) Confidence

The Slope Model achieved 100% accuracy, meaning all 104 test samples were correctly classified.

- **Mean Confidence (Correct):** 26.78%

- **Distribution Analysis:** Despite the perfect accuracy, the confidence scores are generally lower compared to the multimodal model, ranging from a minimum of **19.83%** to a maximum of **48.64%**. This indicates that while the linear decision boundary perfectly separates the classes, the margins are tighter.

Since no errors were made, it was impossible to calculate an optimal rejection threshold based on misclassifications. Figure 2 shows the confidence distribution for the Slope Model.

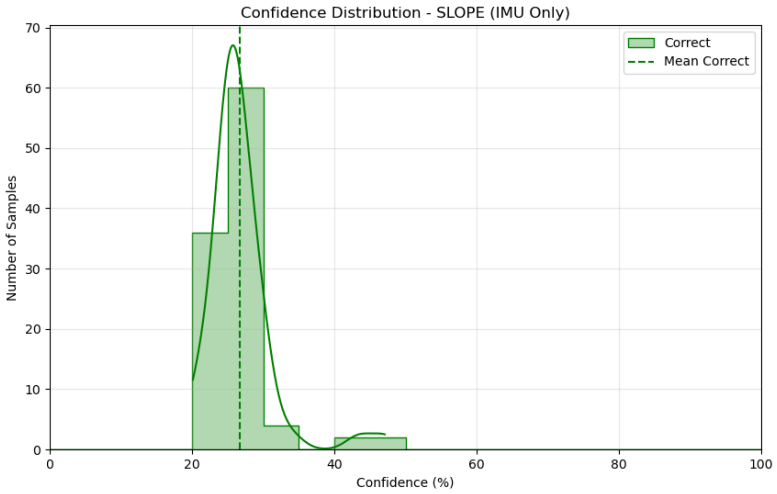

Figure 2: Confidence Distribution for Slope Model (IMU Only)

7 Conclusion

This project successfully demonstrated the design, implementation, and validation of an end-to-end **Multimodal Gait Recognition System**. Starting from a raw and heterogeneous dataset provided by the VisionLab, our primary contribution lay in the robust **Data Engineering pipeline** constructed to sanitize, synchronize, and structure the information.

The transition from unstructured ROS bags and fragmented CSV logs to a coherent dataset was not merely a preparatory step but a critical engineering challenge. The implementation of automated health audits and hash-based leakage prevention protocols ensured that the subsequent machine learning models were trained on scientifically valid, corruption-free data.

7.1 Key Findings

Our experimental results highlight three major conclusions:

1. **IMU Dominance and Multimodality:** For the current dataset and feature engineering, the IMU modality alone proved exceptionally powerful, achieving 100% accuracy for both Walk/Stairs and Slope tasks. While the full multimodal fusion for Walk/Stairs showed a slight "feature dilution" effect compared to IMU-only (99.04% vs 100%), this does not negate the general principle of multimodality's value. In more challenging or noisy real-world scenarios, multimodal systems typically offer enhanced robustness by compensating for the weaknesses of individual sensors. The exceptionally clean IMU data in this specific dataset led to this interesting observation.

2. **Dimensionality Reduction Efficacy:** Despite the "Curse of Dimensionality" introduced by concatenating dense video vectors with inertial data (∼54k features), the **PCA-SVM pipeline** proved highly effective. By retaining 95% of variance, we successfully distilled the unique biometric "Gait Signature" of each subject into a compact latent space, enabling high classification performance with a computationally efficient model.

3. **Security Viability:** The probabilistic analysis demonstrated that the system is not just a classifier but a viable security tool. By establishing a confidence threshold (experimentally set at **26.6%** for the Main Model), the system can reliably distinguish between known personnel and "Unknown" intruders, a crucial feature for real-world access control applications.

8 Real-World Applications and Demonstrators

To validate the practical viability of our models beyond static metrics, we developed two real-time software demonstrators. These applications bridge the gap between the theoretical machine learning pipeline and a deployable biometric security system.

8.1 Gait Security Pro (Access Control System)

The script `predict_demo_security.py` simulates a live access control checkpoint. It integrates the trained SVM model with a decision logic layer derived from our confidence analysis.

- **Logic:** The system processes an incoming gait sequence and predicts the identity. It applies the experimentally determined **Security Threshold (23.6%)**.
  - If $P(\text{class}) >$ Threshold: **ACCESS GRANTED** (Subject Identified).

  - If $P(\text{class}) < \text{Threshold}$: **UNKNOWN SUBJECT** (Access Denied).
  - Internal Sanity Check: If the probability-based prediction conflicts with the strict geometric decision boundary, the system raises a **SYSTEM CONFLICT** alert.

- **Augmented Reality Visualization:** To aid security operators, the interface implements several computer vision filters in real-time:

  1. **Motion Intensity Heatmap:** Accumulates frame differences using a Jet colormap to visualize the energy distribution of the movement.
  2. **Cyber Edges:** Applies Canny edge detection $(50, 150)$ combined with a Gaussian blur to highlight the subject's structural contours with a "glow" effect.
  3. **Lagrangian Particle Tracking:** Uses Sparse Optical Flow (Lucas-Kanade) to draw persistent green traces of key-points over time, visualizing the swing trajectory of limbs.
  4. **Vibrant Depth:** Remaps the normalized depth stream to a Turbo colormap for enhanced depth perception.

8.2 Gait Visualizer (Forensic Analysis Tool)

The script `predict_visual.py` serves as an "Explainable AI" (XAI) tool, designed for forensic analysis of the model's decisions.

- **Comparative Inference:** Unlike the security demo, this tool retrieves the "Canonical" reference data (Run 1) of the predicted subject from the training set and displays it side-by-side with the current test sample.

- **Gait DNA Matrix:** The system generates a comprehensive matrix visualization comparing the biometric features of the Test subject vs. the Reference subject across three dimensions:

  1. **Energy (GOFI):** Comparing the accumulated motion magnitude.
  2. **Silhouette (Mask):** Comparing the static body shape and posture.
  3. **Skeleton (Trace):** Comparing the dynamic temporal evolution of feature points.

This tool allows us to visually verify *why* the model made a specific prediction, confirming that the classification is based on genuine gait kinematics rather than background artifacts.

9 Future Developments

While the current system achieves high performance on the controlled dataset, several avenues for future research and optimization remain:

- **Deep Learning Integration:** The current feature extraction relies on handcrafted Computer Vision (Optical Flow). A natural evolution would be replacing the SVM with a **3D Convolutional Neural Network (3D-CNN)** or a **GaitSet** architecture to learn spatiotemporal features directly from the raw frames end-to-end.
- **Real-Time Optimization:** Currently, the Optical Flow computation (Farneback and Lucas-Kanade) is computationally intensive. Migrating the `gait_processing.py` pipeline to GPU-accelerated OpenCV (CUDA) or using lighter flow estimation networks (e.g., FlowNet) would enable real-time inference at 60 FPS.
- **Covariate Robustness (Clothing and Carrying):** The current model assumes standard clothing. Future work should focus on the specific challenges of frontal recognition, such as subjects wearing heavy coats (which mask limb movement) or carrying objects/backpacks (which alter the center of mass and occlude the silhouette). Collecting a dataset with these "covariates" would be the next logical step to ensure real-world reliability.
- **Longitudinal Stability:** Validating the biometric signature over longer time spans (e.g., weeks or months) to determine if the "Gait Signature" remains consistent for the same subject despite changes in footwear or minor physical variations.

Project Deliverables and Codebase

The project resulted in a modular, reproducible Python codebase organized into five logical subsystems. Below is the detailed breakdown of the developed scripts and their specific functions within the pipeline.

1. Data Engineering & ETL Pipeline

These scripts handle the ingestion of raw heterogeneous data (ROS bags, ZIP archives) and their transformation into the canonical multimodal dataset.

`convert_bags.py` The primary video extractor. It parses binary `.bag` files, decodes synchronized RGB-D-IR streams, applies depth normalization (clipping at 15m), and exports standardized `.avi` files at 60 FPS.

`organize_imu.py` Manages inertial data ingestion. It scans nested ZIP archives, resolves subject naming conflicts (e.g., merging fragmented sessions), and maps raw sensor logs to the canonical action hierarchy.

`multimodal_feature_extractor.py` The bridge between raw data and ML. It orchestrates the feature extraction process, fusing the visual vectors (from `gait_processing`) with the statistical IMU descriptors into the final `.npy` arrays used for training.

2. Quality Assurance & Integrity Audits

A comprehensive suite of validation scripts developed to ensure dataset health, mathematical consistency, and to prove the absence of Data Leakage.

`audit_video_health.py` Performs automated sanity checks on the generated videos, flagging corrupted files or samples with insufficient frame count/file size.

`verify_dataset_completeness.py` It compares the source file index against the destination dataset structure to ensure 100% data migration reliability.

`check_duplicates.py` / `check_video_duplicates.py` Critical validation tools that calculate MD5/SHA256 hashes of features and raw videos to verify zero physical overlap between Training and Test sets.

`check_basename.py` Validates the logical consistency of the Train/Test split. It parses filenames to ensure that the split logic (e.g., Run 5-6 for Test) is applied correctly and that no file is logically assigned to both sets.

`check_dims.py` Verifies the structural integrity of the processed feature vectors. It confirms that the total vector length minus the IMU component matches the expected visual feature dimension (Global Video Len: 24,576), ensuring no features were truncated during extraction.

3. Computer Vision Core

The engine responsible for signal processing and visual feature extraction.

`gait_processing.py` The core library. It implements the static background estimation (median filter), Dense Optical Flow (Farneback algorithm) for GOFI generation, and Sparse Optical Flow (Lucas-Kanade) for skeletal trajectory tracking.

4. Machine Learning & Analytics

Scripts dedicated to model training, optimization, and scientific evaluation.

`train_walk_stairs.py` Trains the Main Multimodal Model. Implements the `StandardScaler` $\rightarrow$ `PCA(0.95)` $\rightarrow$ `Linear SVM` pipeline for Walk and Stairs tasks.

`train_slope.py` Trains the specialized Slope Model (IMU-Only), optimizing the pipeline for low-dimensional inertial features.

`ablation_study.py` Performs comparative experiments (Video vs. IMU vs. Fusion) to quantify the contribution of each modality.

`analyze_confidence.py` Calibrates model probabilities to determine the optimal security rejection threshold (experimentally set at 26.6%).

`visualize_result.py` Generates confusion matrices and performance plots.

5. Real-Time Demonstrators

End-user applications simulating real-world deployment scenarios.

`predict_demo_security.py` **"Gait Security Pro"**: A simulation of an access control checkpoint. It features a Head-Up Display (HUD) with real-time biometric filters (Heatmaps, Cyber Edges) and implements the logic for "Access Granted/Denied" based on confidence thresholds.

`predict_visual.py` **"Gait Visualizer"**: A forensic analysis tool. It compares the test subject against their canonical reference using a "Gait DNA Matrix", visualizing the similarity in Energy (GOFI), Silhouette (Mask), and Kinematics (Trace).

References

[1] H. Ye, T. Sun, and K. Xu, *Gait Recognition Based on Gait Optical Flow Network with Inherent Feature Pyramid*, Applied Sciences, vol. 13, no. 19, p. 10975, 2023.

[2] H. Masood and H. Farooq, *Utilizing Spatio Temporal Gait Pattern and Quadratic SVM for Gait Recognition*, Electronics, vol. 11, no. 15, p. 2386, 2022.

[3] VisionLab, *Computer Vision Laboratory - Department of Computer Science, Sapienza University of Rome*, Official Website, 2025. Available at: `https://visionlab.di.uniroma1.it`.

[4] L. Musso and G. Pietrangeli, *Frontal Gait Flow Recognition - Official Repository*, GitHub, 2025. Available at: `https://github.com/lorenzomussoo/Frontal-Gait-Flow-Recognition.git`