Expert Systems With Applications 269 (2025) 126499

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A scalable gait acquisition and recognition system with angle-enhanced models

Diogo R.M. Bastos [a], João Manuel R.S. Tavares [b,*]

[a] Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal
[b] Departamento de Engenharia Mecânica, Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, s/n, 4200-465 Porto, Portugal

ARTICLE INFO

Keywords:
Biometric
Computer vision
Deep learning
Imaging
Gait recognition
YOLOv8
ByteTrack

ABSTRACT

Person recognition through gait is a highly promising biometric technique, offering substantial advantages over traditional methods. Despite its potential, gait recognition from images can be challenged by factors such as variations in viewing angles, personal accessories, or clothing, which may alter specific gait characteristics. A novel and innovative gait identification system with two main components was developed to address these challenges. The first component focuses on the acquisition of gait sequences, using algorithms for detection, tracking, and gait analysis from images. The gait analysis algorithm facilitates the extraction of high-quality image sequences while determining the subject's movement angle relative to the imaging camera. This is essential for ensuring precise and consistent data for the identification process. The second component is the person identification algorithm, which employs model-free approaches. This component includes various approaches to integrate the angle information into four well-established models: GaitPart, GaitSet, GaitGL, and GaitBase built using the CASIA-B dataset. The results demonstrated that angle information can refine feature extraction when properly integrated into the model, achieving state-of-the-art results across the four models. The GaitPart, GaitSet, and GaitGL models preferred late-stage angle integration, whereas GaitBase performed better with early-stage integration due to its strong backbone. In the final phase of this study, additional tests were conducted using the modified GaitBase model with angle information on the CASIA-E dataset. These tests confirmed the model's effectiveness and enabled a detailed analysis of the threshold that differentiates gait sequences from the same person and those from different individuals. This threshold enhances the system's scalability by enabling it to determine whether a person has been previously observed. Thus, this study developed an innovative and theoretically scalable system adaptable to a growing number of users and locations, with potential applications in access control, security monitoring, and attendance management.

1. Introduction

Gait recognition is a rapidly growing research area and a promising biometric technique, offering significant advantages over traditional methods such as face, iris, and fingerprint recognition (Harris et al., 2022). Its key strengths include non-contact and discrete data collection, enabling the acquisition of walking patterns from a distance, even at lower resolutions (Hawas et al., 2019). Additionally, gait patterns are inherently more difficult to replicate than other biometric traits, enhancing security. Recent advancements in sensing technologies and deep learning have further propelled progress, enabling automatic feature extraction and improved performance in gait-based identification models (Khaliluzzaman et al., 2023). However, gait recognition in images faces significant challenges due to variations in viewing angles, i.e., cross-view conditions, changes in clothing, and the use of accessories, which can alter key gait characteristics and reduce the performance of the computational method (Harris et al., 2022; Russel et al., 2021). Developing a robust computational gait recognition model that can effectively address these challenges remains an open research problem (Khaliluzzaman et al., 2023). Motivated by these challenges, new approaches to enhance the performance of computational gait recognition systems are demanded. One crucial aspect often overlooked is the data acquisition process, which is as essential as the used identification model itself. The quality of the acquired gait sequence images directly impacts the model's performance. In some applications, such as "tag and track" operations, it may be preferable to use all available data instead of filtering for the highest quality sequences, as maintaining continuous tracking is prioritized over detailed identification accuracy.

* Corresponding author.
E-mail addresses: up202202425@fe.up.pt (D.R.M. Bastos), tavares@fe.up.pt (J.M.R.S. Tavares).

https://doi.org/10.1016/j.eswa.2025.126499
Received 18 October 2024; Received in revised form 6 December 2024; Accepted 7 January 2025
Available online 11 January 2025

However, ensuring high-quality image sequences is essential in access control, security monitoring, and attendance management applications, as these scenarios typically involve medium to long-term data storage. The data can accumulate noisy image sequences without proper filtering, undermining future recognition accuracy. To address this, this study proposes a novel gait data acquisition algorithm that integrates YOLOv8n (Jocher et al., 2023) for person detection, ByteTrack (Zhang et al., 2022) for tracking and a Sequence Quality Analysis Module (SQAM). This ensures that the sequences used for identification are free of image, i.e., frame, discontinuities, exhibit minimal body occlusion, maintain consistent direction and orientation, and correspond to a person actively walking rather than being stationary. Tracking and analysis are performed only when the person is fully within the camera's field of view, ensuring complete and high-quality gait sequences. Once a minimum acquisition time is met, the displacement angle of the person relative to the camera is computed. The recognition system is enhanced by ensuring sequence quality and using this displacement angle to improve the performance of state-of-the-art models. This study also proposes leveraging this angle as an additional feature, enabling models to handle cross-view conditions better, ultimately boosting the overall recognition accuracy in complex environments.

More specifically, this work, whose developed source code is freely available at https://github.com/diogobastos07/Innovative-Gait-Acquisition-System.git, lies in three main contributions:

1. Creating a scalable and innovative gait acquisition system based on gait imaging analysis represents a fundamental contribution of this work. The system employs YOLOv8n for person detection, which was trained and evaluated on the CrowdHuman dataset (Shao et al., 2018), and ByteTrack for tracking, along with an implemented SQAM to ensure the acquisition of high-quality gait sequences. Additionally, the system computes the displacement angle of the individual being tracked relative to the imaging camera, which can be used in various ways to support different applications and improve identification accuracy.
2. This work introduces the integration of the subject's displacement angle relative to the imaging camera as a feature to enhance feature extraction in gait identification using model-free approaches. A detailed comparative analysis was conducted using the GaitPart (Fan et al., 2020), GaitSet (Chao et al., 2021), GaitGL (Lin et al., 2022) and GaitBase (Fan et al., 2023) models, along with the CASIA-B dataset (Yu et al., 2006), demonstrating how angle integration can be optimized across different architectures. This approach significantly contributes to the advancement of gait recognition and provides valuable insights into methods for incorporating metadata into complex deep-learning models.
3. The final contribution is a comprehensive study to determine a threshold for distinguishing between new and previously observed subjects, addressing the open-set recognition problem. This mechanism ensures the system can effectively handle registered and unregistered individuals, overcoming intra-class diversity and inter-class similarity challenges. By using the GaitBase_In model, which integrates the displacement angle in the GaitBase architecture, in additional tests on the CASIA-E dataset (Song et al., 2023), scalability in terms of the number of profiles/classes it can accommodate is enabled, making the system adaptable to various real-world scenarios.

2. Related works

2.1. Gait sequence acquisition

Many gait acquisition systems focus on 3D data, such as motion capture with markers or multi-camera setups generating 3D models (Kidziński et al., 2020). While these systems provide precise insights into body movement, they are expensive, require significant infrastructure, e.g., specialized cameras, markers, or sensors, and are primarily used in controlled environments for clinical gait analysis, rehabilitation, or locomotion studies (Ripic et al., 2023). Their complexity and need for controlled conditions often limit their broader application in other fields.

In contrast, 2D camera-based systems and computer vision techniques have become a cost-effective and efficient alternative, especially in more general environments like security and access control. For example, Salehian et al. (2019) utilizes detection and tracking algorithms to capture images of pedestrians in a surveillance system. An initial acquisition is performed, aimed at creating databases for re-identification across cameras. Briefly, this initial acquisition starts when a person is first detected and ends when the tracking algorithm no longer recognizes them. This approach, which relies on initial detection and tracking, is adopted in pedestrian detection and recognition systems. However, such articles often focus on training and evaluating models using well-known datasets, where the data is typically manually pre-annotated for recognition tasks, rather than exploring the capture of high-quality gait sequences.

To the best of our knowledge, no gait acquisition system using a single 2D camera explicitly ensures high-quality gait sequences by considering factors such as active gait with minimal occlusion or interruption. Furthermore, no existing systems are known to address the consistency of gait sequences, which could be ensured by maintaining a uniform sequence size, avoiding abrupt changes in direction, and preserving a consistent orientation across frames. Additionally, calculating the displacement angle relative to the camera, which could serve as an additional feature for identification models, remains unexplored. These aspects may be useful for creating reliable datasets that ensure the quality of the sequences. This could enhance identification accuracy, particularly in real-world scenarios where individual behavior tends to be highly unpredictable.

2.2. Gait recognition

Gait feature representation refers to how human body image information is encoded for identification, depending on model-based or model-free approaches.

Model-based approaches extract image features through geometric representations like skeletons or anatomical landmarks. In the domain of gait recognition, various model-based approaches have significantly advanced the field. For example, Xu et al. (2021) propose a Local Graphical Skeleton Descriptor (LGSD) to capture motion patterns from the skeleton, focusing on features such as position, angle, swing, and trajectory. Zhou et al. (2020) employ Graph Convolutional Networks (GCN) to process skeleton sequences structured as graphs, where joints act as nodes. The Gait-D model, developed by Iwashita et al. (2021), combines GCN with Temporal Convolutional Networks (TCN) to extract spatial and temporal features. The work of Zheng et al. (2022) introduces the MAST-GCN model, which adopts a Spatial–Temporal Graph Convolutional Network (ST-GCN). The model includes an Angle Estimator module and a Part-Frame-Importance (PFI) attention mechanism that adapts to varying view angles, emphasizing significant body parts and frame sequences to improve cross-view recognition. In Upadhyay and Gonsalves (2022), the researchers use Recurrent Neural Networks (RNN) to capture temporal dependencies in gait sequences extracted via OpenPose (Cao et al., 2019). The model processes diverse features, including angular trajectories and temporal displacements, employing sequential modeling to maintain robustness across various conditions and movements. Lastly, Cosma et al. (2023) transform skeleton sequences into image representations for processing by Vision Transformers (ViT). The proposal effectively adapts skeleton sequences to ViT encoders by employing square images and bicubic upsampling, demonstrating the transformers' potential to convert complex temporal

sequences into visual inputs for downstream gait recognition tasks. Theoretically, model-based approaches are more robust to handle changes in clothing, the carrying of objects, and occlusions. However, extracting spatial and temporal features is more complex and relies on accurate pose estimation, which can be challenging, especially in low-resolution images.

Model-free approaches in gait recognition concentrate on analyzing the overall movement of the human body without explicit modulation. Typically, silhouettes represent gait features, leading to superior performance in low-resolution images. In these approaches, silhouettes can be used both frame-by-frame and as more compact representations that integrate spatial and temporal characteristics into a single image, such as Gait Energy Images (GEI). GEI and similar representations are often referred to as template-based approaches. For example, the approach suggested by Hawas et al. (2019) builds GEIs from silhouettes and computes Optical Flow (OF) to capture body part movements fed into the recognition network. Sayeed et al. (2022) used a simple ten-layer Convolutional Neural Network (CNN) with GEI to compare the effects of different activation functions. The CNN with LeakyReLU activation function achieved the best results. On the other hand, Wang et al. (2019) developed a Two-Stream Generative Adversarial Network (GAN) to create view-invariant GEI templates by learning from different angles.

Pose-based strategies, typically used alongside template-based methods, employ compact representations to map each key pose, using alignment and segmentation techniques to enhance recognition under varying conditions. For example, Gupta and Chattopadhyay (2020) addressed speed and frame rate variations by generating Active Energy Image (AEI) templates for each key pose set. These templates are processed through an autoencoder for dimensionality reduction and classified using LDA. Gupta and Chattopadhyay (2021) proposed Dynamic Gait Energy Image (DGEI), a pose-based template generated by mapping frames to predefined key poses. A GAN removes covariant conditions, followed by LDA classification, making the approach effective for mitigating variations such as carrying objects.

Template-based strategies offer the advantage of low computational cost. However, most state-of-the-art models now employ frame-by-frame analysis due to considerable advancements in computational power. When using frame-by-frame analysis, methods can be categorized into video-based and set-based approaches. Video-based methods consider all silhouettes in a sequence in order, allowing for better capture of temporal features in addition to spatial ones. The GaitPart and GaitGL models stand out in this category. GaitPart (Fan et al., 2020) uses focal convolutions to extract part-level features from a sequence of silhouettes without relying on pre-defined templates. The model captures short-range motion patterns by dividing the silhouette into parts and employing a micro-motion capture module (MCM), creating a comprehensive gait representation for identification. GaitGL (Lin et al., 2022) leverages 3D convolutions to simultaneously extract global and local features from gait sequences. Its dual-branch structure captures both contextual information and detailed posture changes, enhancing the model's ability to handle variations in body movements without relying on specific templates. In contrast, set-based methods like GaitSet and GaitBase treat a sequence as an unordered collection of silhouettes, emphasizing the appearance of the silhouettes while still preserving some temporal information. GaitSet (Chao et al., 2021) treats silhouette sequences as unordered sets, extracting robust features through a combination of CNNs for frame-level feature extraction and set pooling methods, which utilize a permutation-invariant function. It employs Pyramid Horizontal Pooling (PHP) to efficiently aggregate and capture hierarchical features. GaitBase (Fan et al., 2023) presents a simple yet powerful model-free set-based approach, using a strong backbone, ResNet9 (He et al., 2015), to extract features from silhouette frames. The model employs temporal and horizontal pooling to aggregate spatial and temporal information, treating the frames as a unified structure. Thus, model-free approaches are more robust in handling cross-view conditions. However, the model's performance significantly decreases due to covariant factors such as object carrying or clothing changes.

To the best of our knowledge, no studies explicitly explore the integration of a person's displacement angle relative to the image camera in identification models. This work proposes this approach using two video-based models, GaitPart and GaitGL, and two set-based models, GaitSet and GaitBase.

3. Proposed system

This work developed the foundations of a comprehensive system for identifying individuals based on gait using 2D images. By employing advanced computer vision and machine learning algorithms, the proposed system can simultaneously detect, track, and analyze multiple individuals in real-time, provided that high-altitude imaging data is used. When a gait sequence meets predefined criteria, the displacement angle relative to the imaging camera is calculated. This angle, along with the imaging data, is sent for segmentation and subsequently to the identification algorithm, which is enhanced by integrating the angle value. The resulting vector is compared to previously recorded profiles for identification. The corresponding profile is updated if a match is found; otherwise, a new profile is created.

The proposed system comprises two main components: gait sequence acquisition and gait recognition. The development of the gait sequence acquisition component is detailed along with its specific techniques and algorithms. For gait recognition, specific changes were conducted in four well-known gait recognition models to integrate angle information for improved feature extraction, which are outlined in the following. CUDA 10.7, Python 3.8.5, and Visual Studio Code were used in the implementation. The computational platform was an NVIDIA DGX workstation with four Tesla V100 GPUs, an Intel Xeon CPU, 256 GB of RAM, and Ubuntu 18.04.6 LTS.

3.1. Gait sequence acquisition

Ensuring data consistency throughout the walking sequence is crucial for effective identification, contributing to a higher gait recognition rate. In the proposed approach, data quality is prioritized over quantity, avoiding storing noisy gait data that would hinder accurate identification. Factors that lead to the exclusion of identification-suitable data include:

1. No movement — Lack of significant movement hinders the extraction of distinct gait characteristics;
2. Body parts out of view — If essential body parts, such as legs or feet, are out of the camera's view, gait analysis may be inconsistent;
3. Direction, i.e., orientation, changes — Abrupt changes in movement direction can disrupt precise gait modeling;
4. Frame discontinuity — Gaps or failures in the frame sequence may result in crucial temporal information loss;
5. Occluded body — Frequent obstruction of the individual's body by objects or other people compromises gait acquisition;
6. Short acquisition time — Insufficient time for data capture limits the amount of information available for accurate identification.

The development began training the object detection algorithm. Then, this algorithm was combined with a tracking algorithm, culminating in the Sequence Quality Analysis Module (SQAM) that ends by determining the individual's angle of movement. Each of these steps is detailed in the following sections.

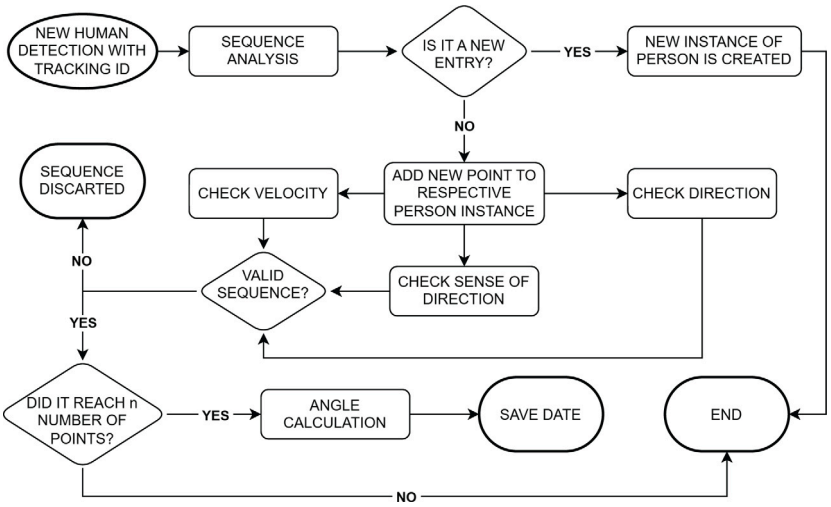

Fig. 1. Workflow chart for the proposed SQAM, illustrating the adopted decision-making process for determining whether a gait sequence is valid.

3.1.1. Object detection

The selected object detection model was YOLOv8 (Jocher et al., 2023), a state-of-the-art computer vision model built by Ultralytics, released in 2023. It represents the latest advancement in the YOLO series by Ultralytics. YOLOv8 was chosen primarily for its compatibility with the project's requirements and for previous studies with the YOLO series. No other detection methods were considered, as the focus was on ensuring a straightforward implementation. YOLOv8 is easy to implement, offers a range of specialized variants for various vision tasks, and supports efficient real-time detection. The available versions are "n", "s", "m", "l", and "x", where "n" is the nano version for resource-constrained environments, and "x" is the extra-large version for maximum accuracy and performance. The nano version was chosen for its speed and lightweight design to test the model in its simplest form. If successful, this confirms that the solution can scale to more robust versions, obtaining results at least as good as those achieved with the nano version.

For the specific requirements of this study, the focus was to ensure that YOLOv8n detected only the full body of individuals. The detections needed to be entirely within the boundaries of the images, excluding cases where parts of the person's body extended beyond the frames. The detection model was configured to ignore people with significant obstructions, ensuring that only individuals with minimal occlusion are detected. This meant avoiding detections where a large body part is blocked or hidden. This was achieved by filtering the annotations of the original dataset selected, as it is explained in Section 4.1.

3.1.2. Tracking

The tracking module uses ByteTrack (Zhang et al., 2022), a multiple object tracking (MOT) algorithm that improves consistency by leveraging high and low-confidence detections. It operates in two stages. First, it matches high-confidence detections to existing tracklets using a higher Intersection Over Union (IoU) threshold, then associates low-confidence ones with unmatched tracklets using a lower IoU threshold for continuity. ByteTrack was integrated with YOLOv8n, using its BB coordinates and class probabilities.

3.1.3. Sequence quality analysis module

The Sequence Quality Analysis Module (SQAM) is organized into core classes, mainly `Person`, `ListPerson`, `LastFrames` and `Diagram`, each serving distinct functions within the system. The `Person` class is responsible for tracking individual people by maintaining their positions, velocities, and other relevant data across image frames, while `ListPerson` manages collections of `Person` objects. The `LastFrames` class stores recent frames for processing, and the `Diagram` class visualizes movement points and angles for performance evaluation.

Table 1
Parameters used in the proposed SQAM.

Parameter	Description	Restrictions
n	Total frames required for the sequence to be considered complete ($n \in \mathbb{N}$).	
p	Minimum number of frames required to start evaluating direction changes ($p \in \mathbb{N}$).	$\begin{cases} 2 \le p < n \\ 1 \le t \\ 2 \le x \le \frac{n}{2} \\ x \times t \le n \end{cases}$
x	Number of consecutive frames considered in speed calculation ($x \in \mathbb{N}$).	
t	Number of frame intervals used to calculate the average speed ($t \in \mathbb{N}$).	
d	Limit of distance allowed between the new point and the trend line ($d \in \mathbb{R}^*_+$).	
v	Minimum limit allowed for average speeds ($v \in \mathbb{R}^*_+$).	

Table 1 outlines the key parameters essential for evaluating gait sequences and their descriptions and associated restrictions to ensure accurate analysis.

The `LastFrames` class stores only potentially necessary frames, with a maximum of n frames, updating as new frames are added to conserve RAM. For each instance of `Person`, only the bounding box (BB) coordinates are saved for sequence quality analysis, and the final cropping of the region of interest (ROI) only takes place after all criteria are met.

The workflow of the SQAM is depicted in Fig. 1. This diagram illustrates the process starting from a new human detection assigned a unique tracking ID by the tracking algorithm. The gait analysis is performed after each new detection is associated with the respective tracklet.

The proposed process (Fig. 1) begins when a new detection is identified, and the system determines if this is a new entry or if it corresponds to an existing person instance. If it is a new entry, a new instance of the `Person` class is created to track that individual. Otherwise, the new point is added to the respective instance of the `Person` class corresponding to the person's tracklet. The algorithm then evaluates the sequence by analyzing three main conditions, ensuring that each data sequence corresponds to a person in motion with a consistent direction and sense of movement. It is discarded if the sequence fails to meet any of these conditions. However, if it satisfies all criteria and reaches n points, i.e., frames, the system calculates the angle of movement relative to the image camera. The validated sequences and their calculated angles are stored for further processing, and the

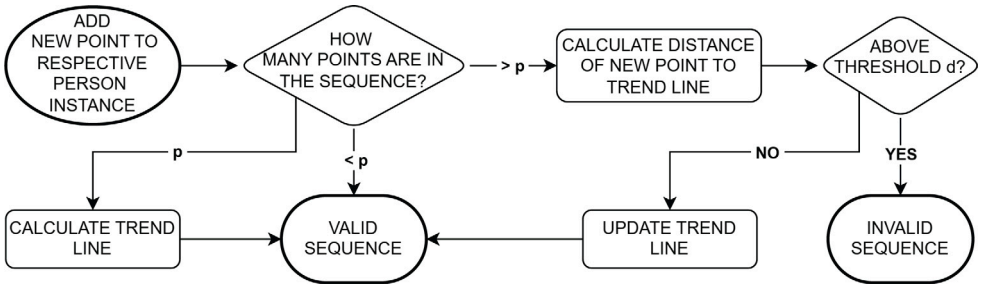

Fig. 2. Workflow for assessing direction changes through trend line evaluation and distance validation.

sequence ends when the data is successfully saved. This process is repeated for each new detection; however, multiple detections can exist in a single frame. In the `ListPerson` class, every tracklet with a newly associated detection is evaluated through a loop that iterates over all tracked detections in the current frame. Once the loop ends, any `Person` objects that did not receive a new detection from the current frame are removed from the list. This ensures that the obtained gait sequences used for identification correspond to individuals consistently detected in every frame of the n total frames.

The diagram of Fig. 2 depicts the detailed process for evaluating changes in direction during movement. The process starts when a new point is added to the respective person instance. First, it checks whether the accumulated points are greater than p. If not, the system lacks sufficient information to calculate a stable trend line. If the sequence has more than p points, the distance of the new point to the trend line is calculated. The next step is to compare this distance with the threshold, d. If the distance exceeds this limit, the sequence is marked as invalid. However, if the distance is within acceptable bounds, the trend line is updated to include the new point, and the sequence is considered valid. The trend line is calculated using Principal Component Analysis (PCA) to identify the primary direction of movement. The distance from a point to the respective trend line is calculated according to:

$$d = \frac{|ax - y + b|}{\sqrt{a^2 + 1}} \tag{1}$$

The diagram of Fig. 3 illustrates how the module calculates the average movement speed and detects direction reversals. The flow begins again by adding a point to the respective person instance. If the total number of points is divisible by x, the speed is calculated based on the last x frames. From this point, the diagram branches into two distinct paths, where invalidating the sequence in either path results in its overall rejection. In one path, the system checks whether the number of points exceeds $2 \times x$. This condition ensures there are sufficient points to evaluate a potential direction reversal. If the signs of the last two speed values, one calculated in the previous step and the other derived from x frames back, are opposite, it indicates a reversal of direction, making the sequence invalid. Otherwise, the sequence is classified as valid. In the second path, the speed is normalized by the BB height. This normalization accounts for the observation that individuals farther from the camera move more slowly while also exhibiting a smaller BB. If the number of points equals or exceeds $t \times x$, it signifies enough data points to assess velocity. The average speed is calculated using the last t speed values, corresponding to $t \times x$ frames for speed evaluation. If the average speed falls below the threshold v, the sequence is invalid due to insufficient speed.

As illustrated in Fig. 4, the angle calculation begins when a sequence reaches n points without failing any criteria. The camera's position is estimated, and a key point is defined at the midpoint of the sequence. Two vectors are then defined: one pointing from the key point to the camera and the other aligned with the trend line of the movement. The angle between these vectors is calculated and adjusted to range from 0° to 360°, providing a comprehensive measure of the person's directional displacement. The angle between the two vectors, $\mathbf{u}$ and $\mathbf{v}$, is computed using:

$$\cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|} \tag{2}$$

3.2. Gait recognition

This section presents the developments proposed to improve the accuracy of cutting-edge algorithms, specifically GaitPart, GaitSet, GaitGL, and GaitBase architectures, by incorporating the displacement angle of the person under tracking relative to the used imaging camera. Thus, the feasibility and benefits of integrating this angle data and the most effective ways to incorporate this information into each architecture are discussed. All integration attempts used the normalized angle value, ranging from 0 (zero) to 1 (one).

3.2.1. GaitPart

The original GaitPart architecture underwent several modifications to incorporate angle information, categorized by their initial, later, and middle integration stages.

The **GaitPart_In** approach added a new channel to each input frame to represent the person's normalized displacement angle, repeating the value to match the dimensions of the silhouette channel. This integration ensured that angle data was captured from the start. For the later integration stage, the **GaitPart_Out** approach, where a 16-dimensional vector was concatenated to each feature vector after the temporal pooling stage (Fig. 5), was implemented. Each vector was generated using separate Fully Connected (FC) layers that received the normalized angle as input. In the **GaitPart_OutLR** variant, a LeakyReLU activation was applied to these vectors before concatenation, adding non-linearity to enhance performance potentially.

The middle stage modifications focused on integrating angle information within the backbone. In **GaitPart_FiLM**, the Feature-wise Linear Modulation (FiLM) (Perez et al., 2017) technique was applied to the output of a Focal Convolution (FConv) layer. FConv divides the channels into s horizontal sections, applying separate convolutions with the same kernel to each section. After LeakyReLU activation, FiLM modulates each section by multiplying the feature values with a trainable γ and adding a trainable β, both scaled by the angle: $f' = \gamma \times f + \beta$. In **GaitPart_FiLM16**, a 16-dimensional vector generated by FC layers using the angle as input replaces the single angle value. The same 16-dimensional vector is applied to all corresponding sections, i.e., the first section in each division uses the same vector. This vector controls γ and β for more detailed modulation across sections. LeakyReLU is applied to the 16-dimensional vectors and the final γ and β. In **GaitPart_NewCh**, an additional channel is added, similarly to the one used in **GaitPart_In**. The specific placement of these techniques within the backbone can be seen in Table 2.

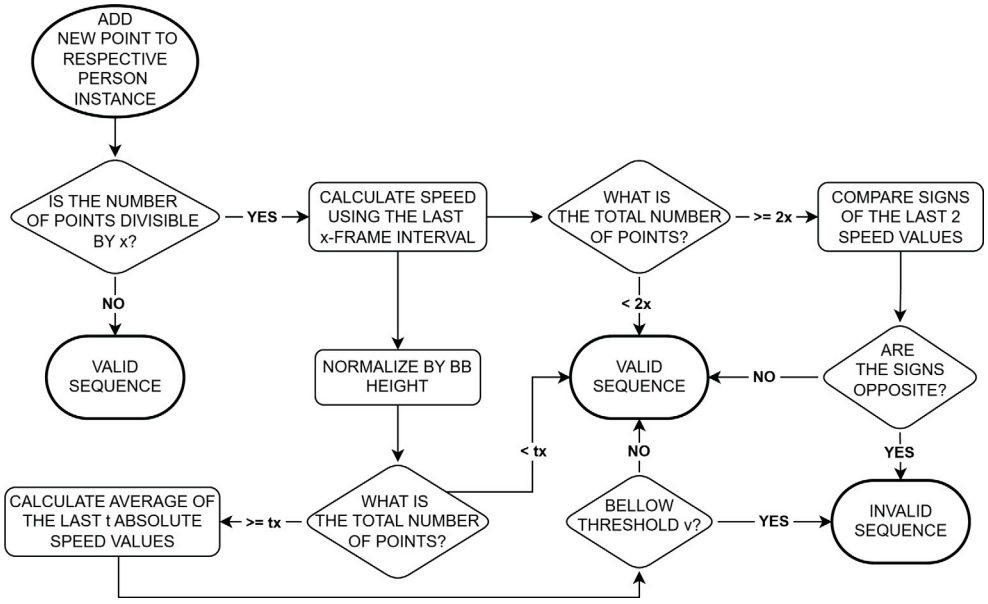

Fig. 3. Workflow for evaluating speed and direction reversal based on calculated speed and threshold checks.

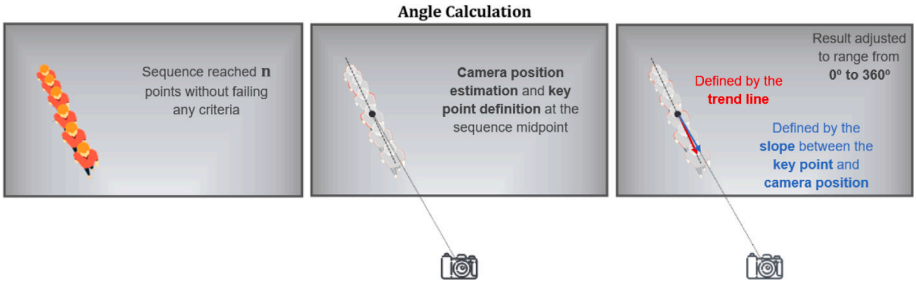

Fig. 4. Workflow for angle calculation, outlining the steps to determine the person's displacement relative to the camera.

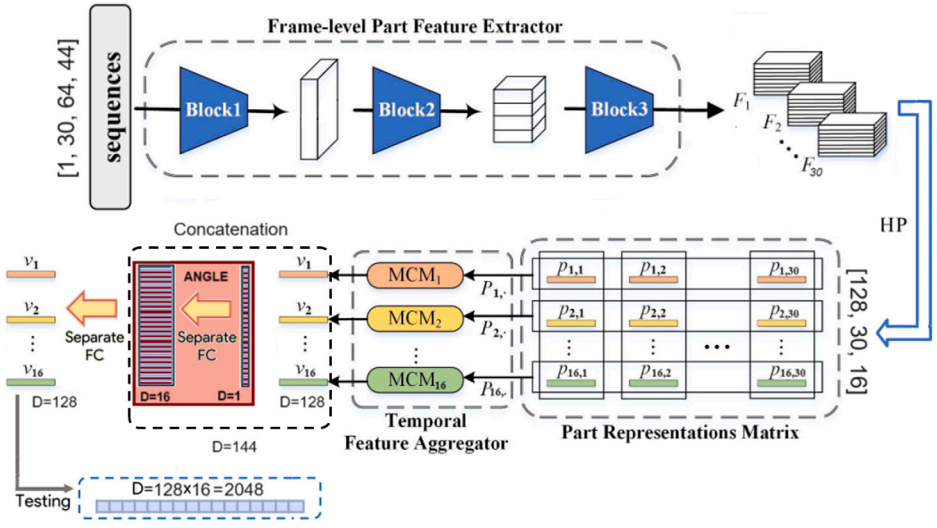

Fig. 5. Pipeline of GaitPart_Out.

3.2.2. GaitSet

The original GaitSet architecture was modified similarly to GaitPart, with changes applied at the initial and final stages. At the initial stage, **GaitSet_In** adds a new angle channel to each silhouette frame, similar to the approach in GaitPart_In. At the final stage, **GaitSet_Out** concatenates 16-dimensional vectors, generated by separated FC using

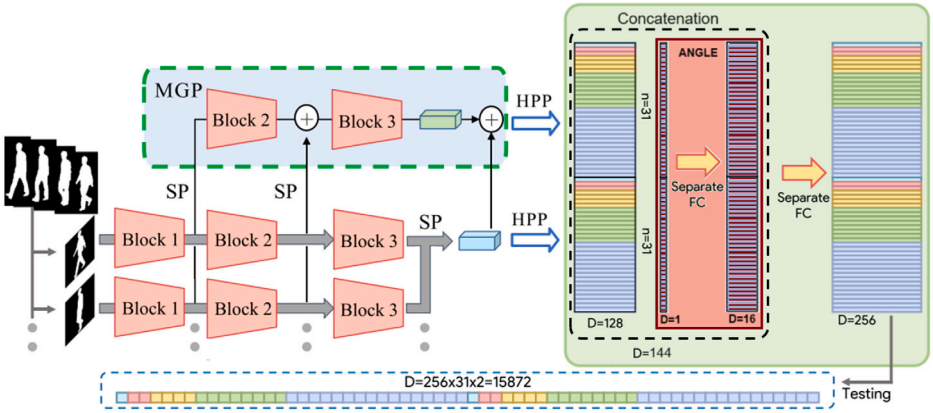

Fig. 6. Pipeline of the GaitSet_Out architecture.

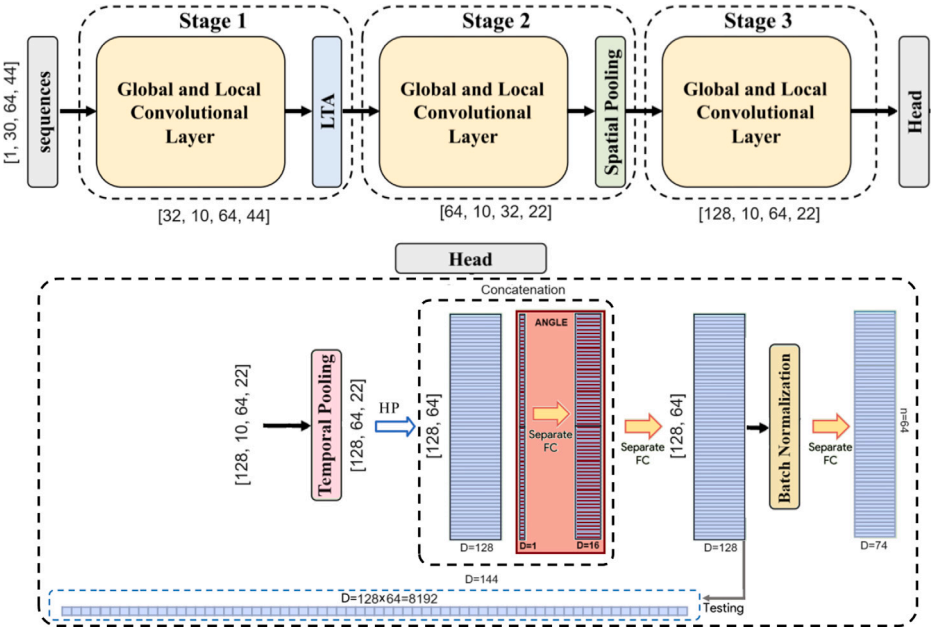

Fig. 7. Pipeline of the GaitGL_Out architecture.

Table 2
Summary of the studied backbone configurations for each model variation (BC — Basic Convolution, M — Max Pooling, FC — Focal Convolution).

Model	Backbone description
GaitPart_FiLM	BC → BC → M → FC → FiLM → FC → M → FC → FC
GaitPart_FiLM16	BC → BC → M → FC → FiLM16 → FC → M → FC → FC
GaitPart_NewCh	BC → BC → M → FC → FC → M → FC → NewCh → FC

the angle as input, to each feature vector resulting from the Horizontal Pyramid Pooling (HPP), as shown in Fig. 6. **GaitSet_OutLR** applies LeakyReLU to each 16-dimensional vector.

3.2.3. GaitGL

Since the GaitGL model processes the entire sequence at once using 3D convolutions, angle integration was only tested at the final stage of the architecture. Similar to the previous modifications, **GaitGL_Out** concatenates 16-dimensional vectors containing angle information to each feature vector after the Horizontal Pooling (HP) stage, as shown in Fig. 7. **GaitGL_OutLR** applies LeakyReLU to each 16-dimensional vector.

3.2.4. GaitBase

In the GaitBase model, the angle integration was explored at the initial and final stages. **GaitBase_In** involved adding a new angle channel to each input frame, while **GaitBase_Out** concatenated 16-dimensional vectors to each feature vector after the Temporal Pooling (TP) stage, Fig. 8. **GaitBase_OutLR** applies LeakyReLU to each 16-dimensional vector.

3.3. Training and test details

3.3.1. Gait sequence acquisition

The YOLOv8n model was trained and evaluated independently from the rest of the gait sequence acquisition algorithm, following the configurations specified in the original documentation. The losses used include Varifocal Loss (cls_loss) designed to address imbalances and uncertainties in classification tasks, CIoU Loss (box_loss) for refined BB regression, and Distribution Focal Loss (dfl_loss) to help the model to estimate object categories better. Training the model involved formatting the dataset to meet the specific requirements and converting the annotations to the correct format.

The trained YOLOv8n model was subsequently combined with ByteTrack and the SQAM components. The integration of the different

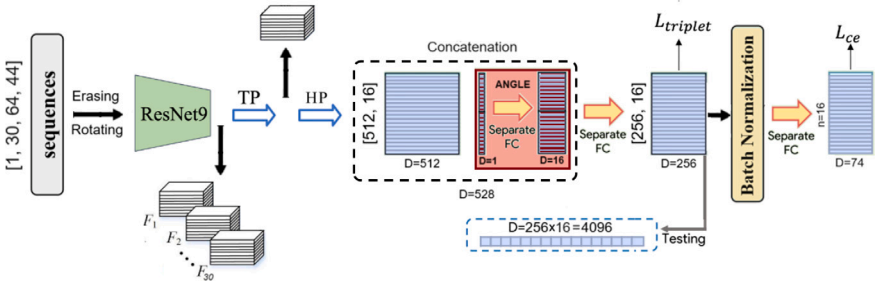

Fig. 8. Pipeline of the GaitBase_Out architecture.

Fig. 9. Frame number 100 from the video used to develop and evaluate the gait data acquisition algorithm.

components of the gait sequence acquisition algorithm was performed and evaluated using an image sequence video with frames as shown in Fig. 9. The used video was obtained from Pexels,[1] which offers various royalty-free videos that can be freely downloaded and used. It was selected for featuring multiple individuals with diverse behaviors, using a high-altitude imaging camera with a wide field of view, ensuring proper tracking and gait analysis. The used video has a frame rate of 25 fps and a duration of 13.6 s, containing a total of 341 frames, with dimensions of 1080 by 1920 pixels. The evaluation was conducted using qualitative analysis. (The used video can be found in the current project's repository at https://github.com/diogobastos07/Innovative-Gait-Acquisition-System.git.)

3.3.2. Gait recognition

The gait recognition algorithms were modified using the OpenGait (Fan et al., 2023) project, a unified and consolidated platform that facilitates a comprehensive study of gait recognition methods. This platform enables the reproduction of the methods developed in this study, with performance comparable to or surpassing the results reported in the original articles. OpenGait also supports the most common datasets, encouraging the research community to use it as a base for developing new solutions. This strategy ensures that researchers follow the same evaluation settings used in the literature, facilitating effective model comparisons.

All modifications made to the studied models were solely aimed at incorporating information about the angle of the person's movement relative to the imaging camera, altering only what was strictly necessary. This ensures that the results obtained solely reflect the contribution of the angle and are not influenced by changes in hyperparameters or other factors.

For the training process, Batch All (BA_+) Triplet Loss (Hermans et al., 2017) was used on the GaitPart and GaitSet models. For the GaitGL and GaitBase model, the sum of the BA_+ Triplet Loss with the Cross-Entropy Loss was used. The batch size is defined by two values, (P, K), where P represents the number of people, and K is the number of gait sequences per person. Both in training and testing, all gait sequences contained exactly 30 frames. In the GaitPart and GaitGL models, the 30 frames used were in order, but the intervals of frames were selected randomly from all available frames. In the GaitSet and GaitBase models, the 30 frames were randomly selected from each sequence.

Test sequences were processed to generate feature vectors for evaluation, the same ones used for the BA_+ Triplet Loss during training. These vectors were split into gallery (labeled) and probe sets. Euclidean distances were computed between probe and gallery vectors, with the closest matches determining the recognition accuracy.

4. Experiments

4.1. Datasets and implementation details

In this study, YOLOv8n was trained and evaluated using the CrowdHuman (Shao et al., 2018) dataset. The various modifications of the gait recognition models were trained and tested on the CASIA-B (Yu et al., 2006) dataset, where the performance comparison was primarily conducted. Based on these results, one selected model was also trained and evaluated on the CASIA-E (Song et al., 2023) dataset to provide a broader assessment. However, conducting a comprehensive performance evaluation of all models on the CASIA-E dataset was not feasible within the available time frame. The time required to train and evaluate a model on CASIA-E is significantly higher compared to CASIA-B due to the greater size and complexity of the CASIA-E dataset.

4.1.1. YOLOv8n

CrowdHuman. The CrowdHuman (Shao et al., 2018) dataset is a large and richly annotated human detection dataset designed to improve the evaluation of detectors in crowded scenarios. It contains 15,000 images for training, 4370 for validation, and 5000 for testing, all collected from the Internet. The training subset includes approximately 340,000 person annotations, and 99,000 ignore region annotations, making it one of the most comprehensive datasets. The dataset features 470,000 human instances in the training and validation subsets, with an average of 22.6 persons per image, capturing various occlusions and crowd scenarios. Each human instance is annotated with three types of BB: head bounding-box (Head BB), human visible-region bounding-box (Visible BB), and human full-body bounding-box (Full BB). The annotations were originally in JSON format, specifying BB as $[x\_min, y\_min, width, height]$.

Implementation Details. This dataset needed to be converted into a required format suitable for YOLOv8n, which involved structuring it into designated folders for images and labels. Each image had a corresponding label file in TXT format detailing class and normalized BB coordinates. The annotations were further filtered to meet the requirements of the system. Firstly, the focus was solely on Full BB. After, Full BB that extended beyond image boundaries and those that

[1] https://www.pexels.com.

lacked sufficient visibility, using a threshold to ensure that the Visible BB area represented at least 70% of the Full BB area. were excluded. To train, the default hyperparameters were used and the number of epochs was defined as 100 and the batch size to 64. A GPU was used to enhance performance. For the model, pre-trained weights provided by Ultralytics, which were obtained with COCO dataset (Lin et al., 2015) that comprises 80 classes, including the "person" class, were used. The complete training of YOLOv8n, including evaluating the validation set at the end of each epoch, took approximately one hour.

4.1.2. Gait recognition

CASIA-B. The CASIA-B (Yu et al., 2006) dataset is one of the most commonly used gait datasets. It features 124 subjects acquired from 11 different view angles (0° to 180°, with 18° intervals). Each angle has 10 gait silhouette sequences under three conditions: normal walking (NM), walking with a bag (BG), and walking with a coat (CL). Specifically, it includes 6 NM sequences, 2 BG, and 2 CL per view, resulting in 13,640 sequences across the dataset, acquired at a rate of 25 fps. For evaluation, the protocol recommended by the OpenGait project was followed. The first 74 individuals were used for training, while the remaining 50 individuals comprised the test set. The test set was further divided into gallery and probe sets. The gallery included the first four sequences of normal walking conditions (NM1–NM4), while the probe set consisted of the remaining sequences: two normal (NM5, NM6), two carrying objects (BG1, BG2), and two with different clothing conditions (CL1, CL2).

CASIA-E. The CASIA-E (Song et al., 2023) dataset stands out as the most recent and potentially the most comprehensive among the gait datasets. It comprises gait silhouette data from 1014 subjects, featuring a total of 384 types of variants for each individual, with two sequences acquired for each variant. The sequences were acquired across three scenarios of increasing complexity: Scene#1, Scene#2, and Scene#3. For each scenario, sequences were acquired under three conditions: normal walking (NM), carrying an object (BG), and wearing different clothing (CL). Each condition has two sets of sequences. Each set contains one sequence for each angle and view. The gait was captured from a horizontal view (L) and a vertical view (H), with 13 distinct angles ranging from 0° to 180° with 15° intervals, along with two repeated angles (45° and 135°) and an angle of 270°. In Scene#3, two additional sets of sequences were acquired for each condition (NM, BG, CL), where the subject pauses and then resumes walking. Following the protocol suggested by OpenGait, the first 200 subjects were used for training, and the remaining 814 were used for testing. The test set was further divided into gallery and probe. The gallery contains the two sets of sequences with NM conditions acquired in Scene#1. The probe includes all sequences acquired in Scene#2 and Scene#3. This includes two sets of sequences for each condition (NM, BG, CL) in each scene and two additional sets of sequences for each condition where the subject pauses and then resumes walking. All sequences acquired at the 270° angle were removed for the test set. For a different evaluation, all sequences acquired from a horizontal view were excluded from the test set to align with the system's design, tailored for gait sequences captured from a vertical view. Sequences with stop-and-resume walking were also excluded for the same reason.

Table 3 summarizes the key differences in complexity between the CASIA-B and CASIA-E datasets.

Implementation Details. The view angles were normalized by dividing the respective value by 180°. Preprocessing included normalizing pixel values to a range from 0 (zero) to 1 (one) and cropping the silhouettes, reducing the original 64 × 64 size to 64 × 44 to remove irrelevant information. For the CASIA-B dataset, rotation and random erasing were also applied to the GaitBase model inputs. These additional preprocessing techniques, with their respective parameters, followed the default conditions established by the OpenGait framework. Training without these augmentations showed overfitting, with the model rapidly achieving perfect training accuracy. The CASIA-E dataset did not require these preprocessing techniques, as experiments without rotation and erasing showed stable generalization. This is due to the dataset's inherent complexity, which provided enough diversity to prevent overfitting without additional augmentations. The training details using the CASIA-B dataset are summarized in Table 4, with the only difference for CASIA-E being a batch size of $(8, 32)$ for the GaitBase model. All models were trained and evaluated using two GPUs in parallel.

Table 3
Summary of the CASIA-B and CASIA-E datasets.

Dataset	#Subjects	#Sequences	#Views	Environment	Other factors
CASIA-B	124	13,640	11	Static indoor	Bag carrying, dressing
CASIA-E	1014	778,752	26	Multiple outdoor	Bag carrying, walking style, dressing, soft biometric features

4.2. Gait sequence acquisition evaluation

4.2.1. YOLOv8n

The trained YOLOv8n model has 168 layers, 3,005,843 parameters, and a computational complexity of 8.1 GFLOPs. As shown in Fig. 10, the downward trends in the loss metrics indicate effective learning and convergence. The achieved performance metric values were 0.79 for precision, 0.69 for recall, 0.78 for mAP50, and 0.52 for mAP50-95. These values indicate high precision minimizing false positives, and a reasonably good recall, ensuring most people are detected. The high mAP50 value reflects accurate BB predictions. The lower mAP50-95 score (0.52) likely stems from the dataset's high density of people, leading to cases where multiple individuals are grouped into a single detection, impacting precision at stricter IoU thresholds.

The obtained performance metric values indicate strong overall performance and reveal sufficient capability even with the simplest version of YOLOv8, the nano version. As seen in the examples in Fig. 11, the model performed well, correctly filtering out annotations of individuals outside the image boundaries or those significantly occluded. The model produced 21% false positives and 31% false negatives, reflecting the expected ambiguity in cases where individuals approach the 70% occlusion threshold, influenced by the inherent uncertainty of manual annotations. Additionally, distant individuals with barely visible silhouettes seem to account for a portion of the false negatives, which is not problematic, as the system is not designed to identify such cases.

Improvements can be made by constructing a dataset with annotations specifically tailored to the system's requirements.

4.2.2. YOLOv8n + ByteTrack

Three representative frames were selected to demonstrate the performance of the ByteTrack model, Fig. 12.

To mitigate the impact of anomalous detections on tracking, only BB with a minimum confidence of 55% was used. Increasing this threshold reduces the number of tracked individuals while lowering it increases false detections or merges multiple individuals in cases of partial overlap. This threshold effectively balances minimizing the impact of anomalous detections while preserving sufficient data for identification.

Considering factors such as the stability of object IDs, the precision of BB, and the algorithm's ability to track different individuals despite variations in movement and appearance, it can be concluded that the algorithm successfully maintains the continuity and accuracy of the tracks over time.

Table 4
Training parameters in CASIA-B: batch size (following the manner introduced in Section 3.3.2), iterations, optimizer (name, momentum and weight decay), margin for the BA_+ Triplet Loss, and learning rate schedule. (The parameters are organized to show the differences in the implementation of each model.)

Model	Batch size	Iter.	Optimizer			Margin (BA_+ TL)	LR schedule
			Name	Mom.	WD		
GaitPart	(8, 16)	120K	ADAM	0.9	0	0.2	Initial 1e−4, decreases to 1e−5 at 100K
GaitSet	(8, 16)	40K	SGD	0.9	5e−4	0.2	Initial 0.1, decreases by 1/10 at 10K, 20K, 30K
GaitGL	(8, 8)	80K	ADAM	0.9	5e−4	0.2	Initial 1e−4, decreases to 1e−5 at 70K
GaitBase	(8, 16)	60K	SGD	0.9	5e−4	0.2	Initial 0.1, decreases by 1/10 at 20K, 40K, 50K

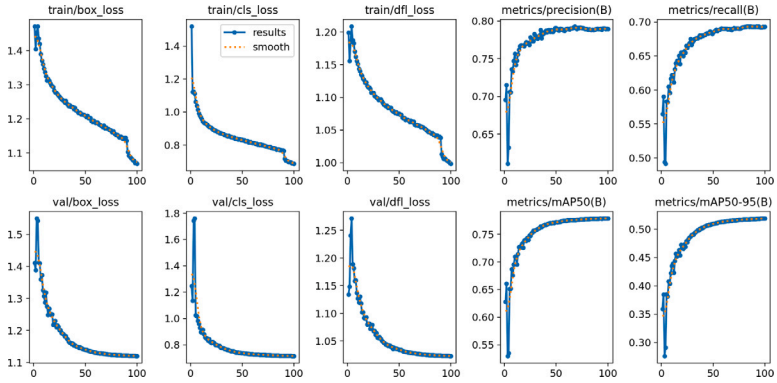

Fig. 10. Training (train) and validation (val) performance metric values obtained for the YOLOv8n object detection model.

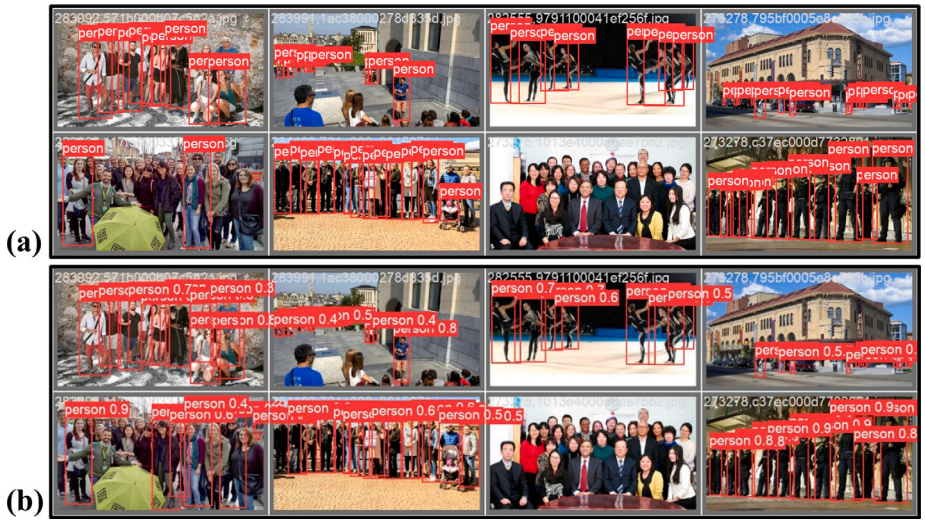

Fig. 11. Comparison of ground truth labels and model predictions for a sample set of 8 images from the CrowdHuman validation dataset: (a) Ground truth labels and (b) Model predictions.

Fig. 12. Visual representation of ByteTrack's performance across different video stages: frames 65, 100, and 135, respectively.

4.2.3. YOLOv8n + ByteTrack + SQAM

During the tuning of the SQAM model, its hyperparameter values were optimized, and the ones in Table 5 were obtained.

The values in Table 5 indicate that a sequence is complete with 75 frames ($n = 75$), corresponding to 3 s. The initial trend line calculation requires a minimum of 10 frames ($p = 10$) to evaluate direction changes. Speed calculations utilize 5 consecutive frames ($x = 5$), while

Table 5
Values of the hyperparameters used in the SQAM model.

n	p	x	t	d	v	Camera position
75	10	5	3	15	0.025	[960, 2000]

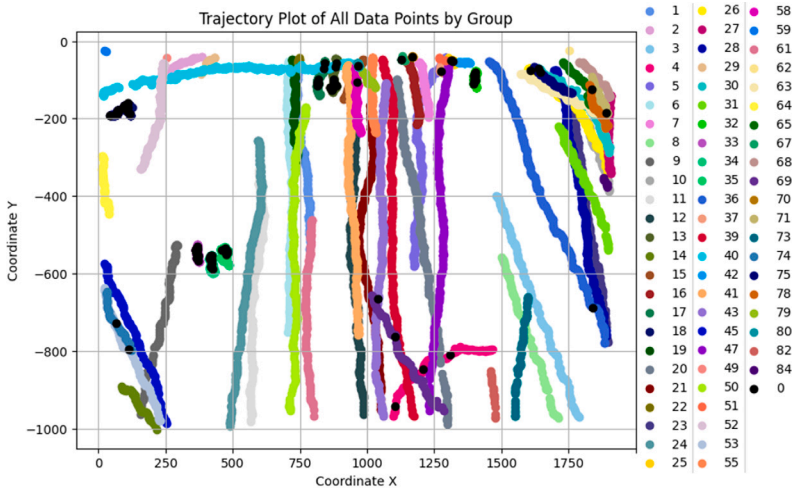

Fig. 13. Visual representation of all data points collected, with colors indicating distinct IDs (black points (0) signify a failure to fulfill the criteria. Y-coordinate values are negative to mimic video frame visualization).

average speed calculations rely on 3 frame intervals ($t = 3$). The distance threshold for validating the proximity of new points to the trend line is set at 15 units ($d = 15$), and the minimum average speed threshold is 0.025 units ($v = 0.025$). Finally, the imaging camera is positioned at coordinates [960, 2000].

Fig. 13 presents the first of two plots generated by the `Diagram` class, illustrating tracked detections marked with their corresponding IDs from the tracking algorithm. The black points indicate moments when sequences were excluded during analysis. Areas with a higher concentration of black points signify individuals or groups that are stationary or moving minimally, confirmed by video footage. Conversely, the sequence with ID#4, located around [1250, −800], demonstrates notable direction changes, leading to three exclusions. Each identified black point prompts the exclusion of the sequence, but a new acquisition begins with the subsequent frame. A 75-frame interval cannot be achieved throughout this gait sequence where all criteria are met, as expected.

Some sequences (Fig. 13) exhibit slight changes in movement direction that would typically prompt exclusion and restart analysis, yet no black point indicates this. The subject with ID#24, near [600, −400], exemplifies this situation. Although the sequence reached 75 frames shortly before, allowing data to be saved, the next 10 frames required for evaluation diminish the significance of the observed movement change. This can be confirmed with Fig. 14, which shows a second plot where black points denote moments when sequences of 75 frames were saved, each associated with a displacement angle relative to the imaging camera.

A trajectory directly towards the camera corresponds to 0°, with angles varying up to 360°, following the clockwise direction. The angle calculation strategy accounts for how the person of the person under tracking affects the perceived movement direction. For example, the trajectory of subject ID#45, shown in the lower-left corner of Fig. 14, has an angle of 1°, indicating movement directly towards the camera, as confirmed in Fig. 15. The accuracy of the angle calculation can be qualitatively confirmed not only for the subject ID#45 but also for all other subjects.

Many of the sequences in Fig. 13 do not show black points because they did not fail any of the established criteria. However, they also do not appear in Fig. 14 because they did not reach the 75 frames needed.

The number of frames can be reduced to as low as 15, considering the restrictions outlined in Table 1. However, it is reasonable to assume that longer sequences improve the accuracy of future identification results. Additionally, SQAM effectively filters out false positives and negatives, as they are unlikely to form a valid gait pattern across multiple frames.

The analysis of the processing times for the acquisition component revealed an average per frame of 41.95 ms for inference, 6.19 ms for tracking, and 8.30 ms for SQAM when using the CPU. Further optimization is necessary for compatibility with 25 fps videos, which requires processing times of less than 40 ms per frame. An effective solution is to transform the model for TensorRT use, as highlighted in the official YOLO documentation, to exploit GPU capabilities fully. The documentation reports that tests with YOLOv8n, a model comprising 3.2 M parameters and 8.7 GFLOPs, achieved inference times of just 0.99 ms per frame on an A100 GPU with TensorRT. Given that the suggested model is slightly smaller, with 3.01 M parameters and 8.1 GFLOPs, even faster processing times are anticipated, ensuring the feasibility of efficiently meeting the required time constraints.

While the SQAM parameters were optimized to balance accuracy and computational efficiency, a more detailed exploration of their effects remains an open question. The influence of the parameters in the SQAM is a topic of significant interest, as it directly impacts the quality of the acquired gait sequences. Future work will focus on systematically studying the sensitivity of the SQAM parameters to understand their role better and optimize their values for diverse scenarios.

4.3. Performance comparison on CASIA-B

Table 6 highlights the Rank-1 accuracy performance comparing the original models with their versions integrating angle information. The results are divided by different walking conditions, averaged across all probe and gallery views, excluding identical-view cases to simulate cross-view conditions.

In all models, at least one variation showed improved results, demonstrating that incorporating angular information enhances the model's ability to discriminate and generalize across different viewing angles and conditions.

Looking at average accuracies, variations of the GaitPart, GaitSet, and GaitGL models improved upon their original models in all versions integrating angular information at both early (_In) and late stages (_Out, _OutLR), except for GaitSet_In, which only matched the original. Late-stage integration outperformed early integration in GaitPart and GaitSet, with LeakyReLU further boosting performance in GaitPart and GaitGL. Specifically, their best versions, GaitPart_OutLR, GaitSet_Out, and GaitGL_OutLR showed overall accuracy improvements of 0.8%, 0.5%, and 0.4%, respectively. In contrast, GaitBase was the only model where late-stage angular integration was not beneficial. However, GaitBase_In achieved a notable 0.7% overall improvement, higher than any other early-stage integration.

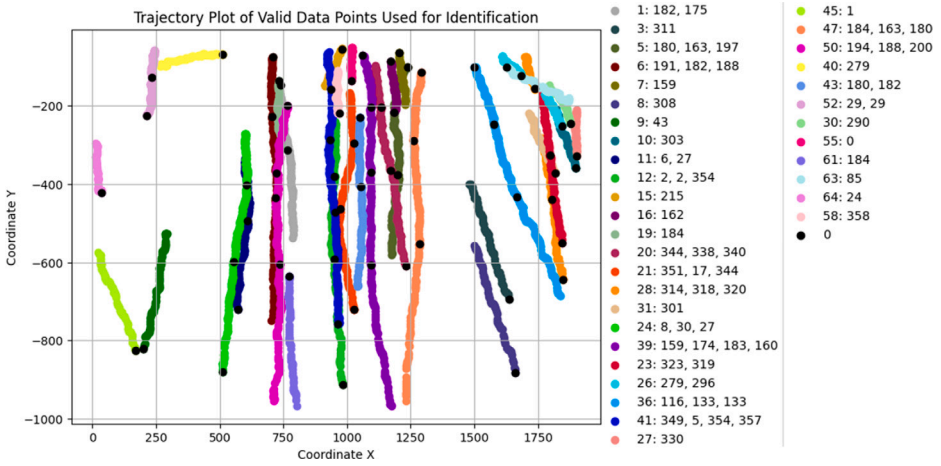

Fig. 14. Visual representation of data points meeting all criteria for identification, with each group shown in a unique color (black points (0) denote the final stored data point in each valid trajectory, along with the associated displacement angles relative to the imaging camera).

Fig. 15. Raw images of the subject with ID#45: (a) Natural inclination of the person under analysis, and (b) Image rotation to provide a better perception of the movement direction.

Table 6
Rank-1 accuracy comparison of original models vs. angle-integrated versions across walking conditions (NM — normal walking, BG — walking with a bag and CL — walking with a coat, identical-view cases were excluded, best values in bold). For each model, the average percentage improvement of the best-performing angle-integrated version compared to the original model is as follows: GaitPart (+0.8%), GaitSet (+0.5%), GaitGL (+0.4%), and GaitBase (+0.7%).

Model	NM	BG	CL	Average
GaitPart	96.3	90.7	77.4	88.1
GaitPart_In	96.1	90.7	78.6	88.5
GaitPart_Out	96.1	91.1	78.5	88.6
GaitPart_OutLR	**96.7**	**91.2**	**78.9**	**88.9**
GaitPart_FiLM	78.5	70.3	54.9	67.6
GaitPart_FiLM16	89.0	78.7	60.5	76.1
GaitPart_NewCh	95.2	85.2	72.7	84.4
GaitSet	95.8	90.2	74.3	86.8
GaitSet_In	95.8	90.0	74.5	86.8
GaitSet_Out	**96.0**	**90.5**	**75.4**	**87.3**
GaitSet_OutLR	**96.0**	90.2	74.5	86.9
GaitGL	97.3	94.6	83.9	91.9
GaitGL_Out	97.4	94.9	83.7	92.0
GaitGL_OutLR	**97.6**	**95.0**	**84.3**	**92.3**
GaitBase	**98.2**	93.8	77.5	89.8
GaitBase_In	98.1	**94.5**	**79.0**	**90.5**
GaitBase_Out	98.1	94.0	76.9	89.7
GaitBase_OutLR	98.0	93.8	77.3	89.7

The results also show strong performance under clothing conditions (CL1, CL2), with GaitPart_OutLR, GaitSet_Out, GaitGL_OutLR and GaitBase_In achieving improvements of 1.5%, 1.1%, 0.4% and 1.5%, respectively. These findings suggest that when angular information is properly integrated, it enhances the robustness of all models, with particularly significant improvements in more variable and challenging conditions. The results also highlight the varying impact of angle integration, depending on the model and the specific configuration.

None of the studied models with mid-stage angle integration outperformed the original GaitPart in overall performance. The GaitPart_FiLM model experienced a drastic performance drop due to the FiLM layers scaling the feature map to zero at 0°, resulting in significant information loss. While GaitPart_FiLM16 addressed this issue, its overall performance remained poor, suggesting that FiLM layers require better calibration or may not be suitable. Similarly, GaitPart_NewCh failed considerably to enrich contextual learning, indicating that introducing angular data at intermediate stages is inadequate to enhance feature extraction.

Table 7
Comparison of parameter size, training time, testing time, and inference time for the base models and their best angle-integrated variations.

Model	Parameters number (M)	Training time (h)	Testing time (s)	Inference time (ms)
GaitPart	1.20400	8.49	78	14.22
GaitPart_OutLR	1.23702	8.51	79	14.40
GaitSet	2.59459	1.99	23	4.19
GaitSet_Out	2.84954	2.00	24	4.38
GaitGL	3.09667	2.87	33	6.02
GaitGL_OutLR	3.22877	2.90	34	6.20
GaitBase	7.30541	10.89	30	5.46
GaitBase_In	7.30598	11.12	31	5.65

Table 7 demonstrates the computational costs associated with the base models and their best angle-integrated variations. The superior results achieved by the best angle-integrated versions of each base model were accompanied by a slight increase in inference time, remaining below half a millisecond in all cases. This minimal increase is negligible, proving the efficiency of angle integration.

4.3.1. Ablation study

Three studies were conducted: the first two aimed at understanding why integrating angle information at an intermediate stage was ineffective, and the third to explore why variations in the GaitBase model with later-stage integration did not lead to improvements as seen in other models.

Analysis of different hyperparameters on GaitPart_NewCh. Additional tests with GaitPart_NewCh were conducted motivated by the conviction that the poorer results were due to overfitting caused by excessive adaptation to the angle value. Fig. 16 shows three distinct groups of lines, with the mean triplet distance used as the primary

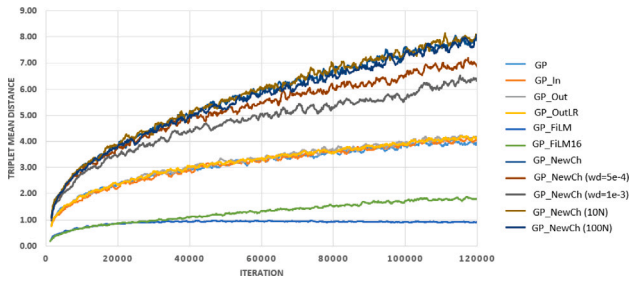

Fig. 16. Mean triplet distance over iterations for different variations of the GaitPart model.

Table 8
Rank-1 accuracy comparison of GaitPart_NewCh with variations in weight decay and added noise to angular channel across walking conditions (NM — normal walking, BG — walking with a bag and CL — walking with a coat, identical-view cases were excluded).

Condition	Value	NM	BG	CL	Average
–	–	95.2	85.2	72.7	84.4
Weight decay	0.0005	95.5	86.1	74.6	85.4
	0.001	95.6	86.2	74.6	85.5
Noise	10%	95.5	85.5	73.7	84.9
	100%	95.2	86.1	73.7	85.0

metric to assess class separation during training. Triplet loss was less informative due to PyTorch's autograd mechanism, which optimizes only the hardest triplets, leading to localized adjustments without noticeable changes in loss values. While the number of hard triplets could also be presented, it provided less visual clarity.

Variations that integrated angle information at early and later stages showed consistent improvement, closely matching the original GaitPart's training behavior, as expected from the good accuracy results. However, variations with intermediate-stage integration performed worse for different reasons. For example, the GaitPart_FiLM and GaitPart_FiLM16 models showed a significant drop in the mean triplet distance, confirming these approaches struggled with maintaining representation quality. In contrast, GaitPart_NewCh had notably higher triplet mean distances, suggesting overfitting and a loss of generalization. Further tests, summarized in Table 8, explored different weight decay values and added noise to the angular channel to address overfitting. Noise was introduced by adding a slight random deviation (normal distribution, standard deviation 0.02) to the normalized angle, followed by replacing a percentage of angular values (selected randomly) with random numbers between 0 (zero) and 1 (one).

Initially, weight decay was set to 0 (zero), but introducing values of 0.0005 and 0.001 reduced overfitting, as shown in Fig. 16. Although performance improved, the gains were modest compared to the base GaitPart model. Similarly, adding 10% of noise to the angular channel provided an insignificant boost in performance. Interestingly, with 100% noise, making the angle information essentially irrelevant, the behavior was similar. This suggests that the root cause of overfitting may not be excessive angle adaptation as initially thought. Instead, angle integration in intermediate layers might act like noise, adding useless information, confusing the model, and exacerbating overfitting.

Analysis of the new channel's position in the GaitPart_NewCh. This study evaluated the impact of integrating a new channel containing angle information at various positions within the GaitPart backbone. The results, summarized in Table 9, revealed a trend where earlier integration of the new channel led to improved performance. Specifically, variations with the new channel positioned earlier in the architecture, such as GaitPart_NewCh3, demonstrated higher accuracy across various conditions. However, all variations still exhibited a considerable decline in performance compared to the original GaitPart model. This trend may support the theory that angle information behaves like noise, where the detrimental impact increases as the channel is added later in the network.

Table 9
Rank-1 accuracy comparison of GaitPart_NewCh with variations in the new channel's position across walking conditions (identical-view cases were excluded, BC — Basic Convolution, M — Max Pooling, FC — Focal Convolution).

Model	NM	BG	CL	Average	Backbone description
GaitPart_NewCh	95.2	85.2	72.7	84.4	BC-BC-M-FC-FC-M-FC-NewCh-FC
GaitPart_NewCh2	95.1	86.9	75.7	85.9	BC-BC-M-FC-NewCh-FC-M-FC-FC
GaitPart_NewCh3	95.7	87.4	76.0	86.4	BC-NewCh-BC-M-FC-FC-M-FC-FC

Table 10
Rank-1 accuracy comparison of additional variations of GaitBase with later-stage angle integration (NM — normal walking, BG — walking with a bag and CL — walking with a coat, identical-view cases were excluded, best values in bold).

Model	NM	BG	CL	Average
GaitBase	98.2	93.8	77.5	**89.8**
GaitBase_Out2	97.9	93.4	77.3	89.5
GaitBase_OutLR2	98.1	93.6	**77.8**	**89.8**
GaitBase_2Out32LR	98.1	**93.9**	77.0	89.7
GaitBase_2Out32LR1	98.2	93.8	77.3	**89.8**
GaitBase_2Out64LR1	98.0	93.7	77.2	89.6
GaitBase_Out64LR1	**98.3**	93.8	76.7	89.6

The values in Table 9 also confirm the challenge of improving results when integrating angle information at intermediate stages. A similar behavior is anticipated for integrating other types of metadata in various other models with different purposes. Early and late integration in deep learning models appear to be the most reliable approaches.

Analysis of additional variations with later stage angle integration in GaitBase. Further modifications were made to the later stage angle integration in the GaitBase model, motivated by the hypothesis that the poor performance of GaitBase_Out and GaitBase_OutLR might be linked to the concatenation of 16-dimensional vectors with the 512-dimensional feature vectors derived from the TP. These high-dimensional vectors contrast the 128-dimensional vectors used for angle information concatenation in the other three models. The modifications made are illustrated in Fig. 17.

GaitBase_Out2 first transforms each of the 512-dimensional vectors into 256-dimensional vectors, to which the 16-dimensional angle vectors are concatenated. GaitBase_OutLR2 includes a LeakyReLU activation function. GaitBase_2Out32LR transforms each 16-dimensional vector into 32-dimensional vectors for concatenation. After concatenation, GaitBase_2Out32LR1 transforms the previous concatenated vectors back into 512-dimensional. GaitBase_2Out64LR1 converts the angle information directly into 32-dimensional vectors, followed by transformation into 64-dimensional vectors for concatenation. The resulting concatenated vectors are again transformed into 512-dimensional vectors. Finally, GaitBase_Out64LR1 performs the same process, but the 64-dimensional vectors are obtained directly from the angle values. All these variations apply LeakyReLU to the vectors carrying angle information. Every transformation from one set of vectors to another was accomplished through separate FC layers.

The additional variations, as shown in Table 10, demonstrated occasional improvements in specific walking conditions, but all models achieved an overall accuracy ranging between 89.5% and 89.8%, with none surpassing the performance of the original GaitBase. This adds an extra layer of surprise to the notable improvement achieved by the GaitBase_In model, suggesting that the backbone of GaitBase is the critical factor driving its success. According to the authors (Fan et al., 2023), ResNet9 plays a crucial role in feature extraction, which seems validated by the results.

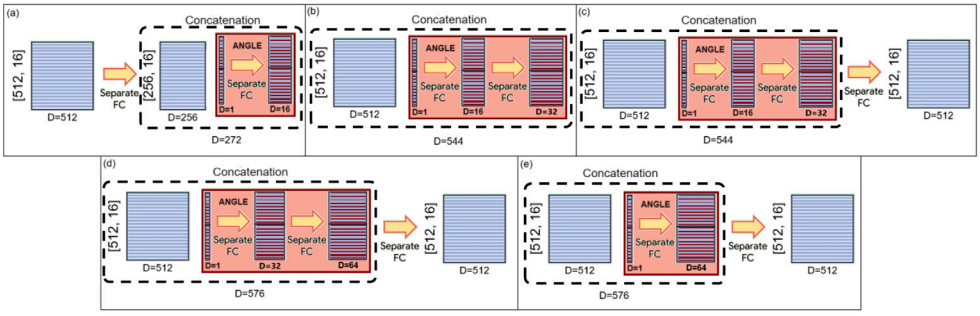

Fig. 17. Different GaitBase modifications to include angle information in later stages: (a) GaitBase_Out2, (b) GaitBase_2Out32LR, (c) GaitBase_2Out32LR1, (d) GaitBase_2Out64LR1, and (e) GaitBase_Out64LR1.

Table 11
Rank-1 accuracy comparison of GaitBase_In and variation across walking conditions on CASIA-E, divided by two evaluation protocols (NM — normal walking, BG — walking with a bag and CL — walking with a coat, identical-view cases were excluded, best values in bold).

Evaluation protocol	Model	NM	BG	CL	Average
Default	GaitBase	91.3	86.3	74.4	84.0
	GaitBase_In	**92.2**	**87.6**	**76.8**	**85.4**
	GaitBase_InN	91.4	86.6	74.7	84.2
Just vertical data	GaitBase	85.1	78.5	66.3	76.6
	GaitBase_In	**86.4**	**80.3**	**69.1**	**78.6**
	GaitBase_InN	85.3	78.9	66.6	76.9

4.4. GaitBase_in evaluation on CASIA-E

GaitBase_In was selected for further evaluation due to its strong performance on the CASIA-B dataset, confirmed by the results in Section 4.3. While GaitGL variations performed better, their evaluation on the CASIA-E dataset posed additional challenges. Unlike other models, which are prepared in the OpenGait repository for fast training and evaluation with pre-defined parameters, GaitGL lacks configurations tailored to CASIA-E, necessitating extensive parameter tuning that was unfeasible given the available time. Additionally, CASIA-E, being a large dataset comparable in scale to OUMVLP (Takemura et al., 2018) and GREW (Xianda et al., 2022), would likely require added layers of complexity, as observed for those datasets. These adjustments increase the model's size to approximately 14.47M parameters and significantly raise the number of iterations needed, often doubling or tripling those used for CASIA-B. Consequently, similar modifications for CASIA-E would result in considerably longer processing times, making evaluation impractical under current constraints. Furthermore, GaitBase has been shown to outperform GaitGL on outdoor datasets, which more closely resemble the real-world scenarios of CASIA-E (Fan et al., 2023).

To evaluate the robustness of GaitBase_In and the effect of noise on angle data, a variant called GaitBase_InN was built. This model adds noise to the angle values during training and evaluation. The noise follows a normal distribution (mean equal to the normalized angle value, standard deviation of 0.02), with adjustments to keep the values between 0 (zero) and 1 (one). Table 11 highlights the Rank-1 accuracy performance by comparing this variation with the original GaitBase. The results are divided by different walking conditions, averaged across all probe and gallery views, excluding identical-view cases to simulate cross-view conditions. The models were evaluated using the default evaluation protocol and a second evaluation focusing on sequences acquired from a vertical view, as explained in Section 4.1.2.

The results in Table 11 highlight the significant improvements achieved by GaitBase_In, particularly due to the integration of angle information. Across the default protocol, GaitBase_In demonstrated an average improvement of 1.4% compared to the baseline GaitBase (85.4% vs. 84.0%). It showed even stronger improvements on CASIA-E than on CASIA-B (1.4% vs. 0.7%), demonstrating how angle integration enhances generalization. Using just vertical data, the improvement was even greater, with an average gain of 2.0% (78.6% vs. 76.6%). Notably, it distinguished gait sequences from 814 individuals despite training on only 200, underscoring its robustness.

Table 12
Comparison of parameter size, training time, testing time, and inference time for the GaitBase and GaitBase_In, using CASIA-E dataset.

Model	Parameters number (M)	Training time (h)	Testing time (s)	Inference time (ms)
GaitBase	7.82150	17.67	4135	6.71
GaitBase_In	7.82208	18.85	4879	7.90

Although the gains for GaitBase_InN were smaller, it still surpassed the baseline GaitBase, indicating the resilience of angle information, even with added noise. Noise was introduced to assess whether the optimal results observed could also be achieved without relying on fixed angle values. During the training of GaitBase_In, the dataset considers sequences captured from 13 different angles, and each sequence is associated with its respective normalized angle value, which can only assume one of these 13 predefined values. By introducing noise, the angle values are no longer fixed, allowing them to vary within a wider range. This experiment also evaluated the model's robustness by verifying whether slight variations in the actual angle would result in significant drops in accuracy. The standard deviation of 0.02, corresponding to an angular variation of approximately 3.6°, was selected arbitrarily as a reasonable starting point. However, further tests with different values could be conducted to study this choice in future work. Based on the results, it is believed that the best approach would be to approximate the angle calculated by the gait acquisition component to the nearest value within the range of angles used during training. Nevertheless, additional tests in real-world contexts are needed to confirm whether this would be the most effective strategy.

Performance dipped slightly for all models in the vertical-view evaluation, likely due to reduced gallery variety, fewer discriminative features, and a mismatch between training and evaluation data. To improve, future work could involve pre-training on the full dataset followed by fine-tuning for vertical-view sequences, which would better adapt the model to this specific scenario.

Table 12 demonstrates the computational costs for GaitBase and GaitBase_In using the CASIA-E dataset. The integration of angle information into GaitBase resulted in a slight increase in inference time, amounting to just 1.19 ms. This minimal increase is negligible, further demonstrating once again the efficiency of angle integration, now also confirmed on the CASIA-E dataset, while still providing performance gains.

A limitation of this study is that training and evaluation were conducted within the same datasets, without testing the models on unseen datasets. While this provides insights into performance within a controlled setting, it does not fully assess robustness in cross-dataset

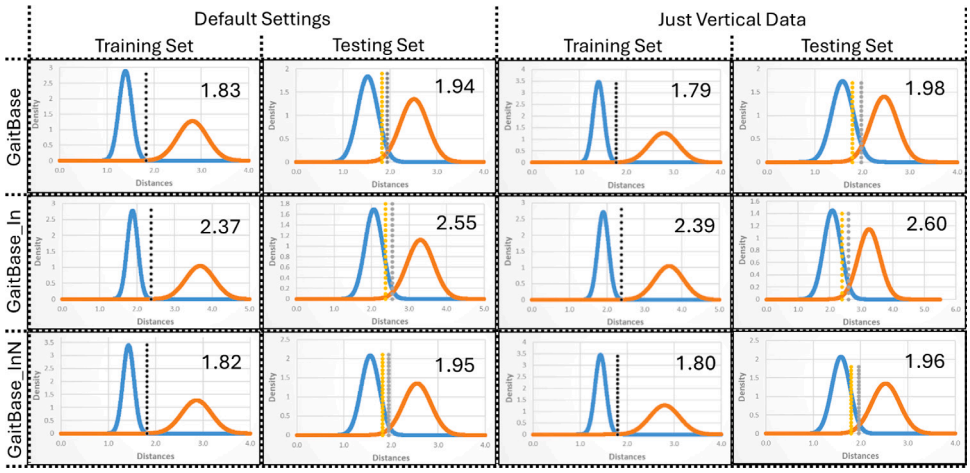

Fig. 18. Intra-class (blue line) and inter-class (orange line) distance distributions for GaitBase, GaitBase_In, and GaitBase_InN divided into training and test sets across the two evaluation protocols (thresholds are marked in black (training) and in gray (testing), indicating their respective values; yellow threshold in the testing sets represents the threshold value calculated in the corresponding training set).

scenarios, which would further approximate the system's real-world performance. Testing in such scenarios would benefit from the availability of more datasets like CASIA-E, which capture gait from elevated perspectives with varying angles. Nevertheless, the current results already provide a solid indication of the model's robustness and capability and the practical impact of angle integration.

4.4.1. Threshold study

This study aimed to determine the optimal threshold that effectively separates intra-class distances, i.e., gait sequences from the same individual, from inter-class distances, i.e., gait sequences from different individuals. Establishing this threshold is crucial for the system's ability to recognize whether a given observation corresponds to a previously observed individual or a completely new one. This capability enhances the system's scalability by accommodating more stored profiles.

The threshold determination was performed for both the default evaluation protocol and the protocol using only vertical view data. The threshold value was established on the training set and evaluated on the test set. For this process, the training set was divided into a gallery and probe set according to the expected evaluation protocol. All distances between the probe and gallery samples were calculated, allowing for the computation of each group's mean and standard deviation from 30,000 sampled distances corresponding to intra-class and inter-class distances. The threshold was defined as the distance where the confidence of belonging to the intra-class distribution equals that of the inter-class distribution, assuming normal distributions. After determining this threshold using the training set, the process was repeated for the test set. A comparison was then made between the thresholds calculated from the training and test sets.

The results shown in Fig. 18 reveal a clear separation between intra-class and inter-class distances, with minimal overlap in the distributions from the training set. This aligns with the goals of the triplet loss function. The test set further reinforces this distinction, highlighting the models' strong performance and generalization capability.

Table 13 presents the obtained accuracy values for recognizing observations as new or previously seen, showing true positive accuracy ("Intra") and true negative accuracy ("Inter") for the three models across three combinations. The values were estimated using a standard normal distribution's Cumulative Distribution Function (CDF). The "Training Set" column displays probabilities derived from the training-set threshold evaluated in the training set, the "Obtained" column presents actual test results using the same threshold evaluated in the testing set, and the "Expected" column reflects the optimal results in the testing set if the threshold had been fine-tuned.

Table 13
Accuracy rates for true positives and true negatives for GaitBase, GaitBase_In, and GaitBase_InN under training, obtained, and expected threshold conditions in two evaluation protocols (best values in bold).

Evaluation protocol	Model	Training set		Obtained		Expected	
		Intra	Inter	Intra	Inter	Intra	Inter
Default	GaitBase	99.92	99.92	**91.96**	98.97	97.19	97.39
	GaitBase_In	**99.97**	**99.97**	90.68	**99.55**	**98.15**	**98.25**
	GaitBase_InN	99.96	99.96	91.38	99.36	97.95	97.97
Just vertical data	GaitBase	99.95	**99.94**	81.60	99.09	95.80	95.45
	GaitBase_In	**99.96**	99.56	**86.85**	99.30	97.01	96.81
	GaitBase_InN	99.92	99.92	86.81	**99.31**	**97.42**	**97.27**

The results in Table 13 correspond to the distribution patterns in Fig. 18. In every case, the obtained threshold is shifted leftward from the expected optimal value, decreasing true positive accuracy and increasing true negative accuracy. While this may initially appear detrimental, it will likely enhance overall system performance. Since inter-class distances greatly outnumber intra-class distances, even a slight reduction in true negative accuracy can lead to a significant number of new individuals being incorrectly associated with previously observed ones. As the number of stored profiles increases, this imbalance becomes more pronounced, underscoring the importance of maintaining a threshold that slightly penalizes true positive accuracy to enhance true negative accuracy. However, if desired, a slight penalty could be applied to the obtained threshold value, bringing it closer to the ideal separation point. This assumes the deviation as an established fact. Moreover, the better results observed with the models incorporating angle information are confirmed by higher accuracy in the "Expected" column, indicating that these distributions have a smaller area of overlap.

Considering the default settings, the accuracy of true positives and true negatives shows little distinction among the models. However, GaitBase_In should be emphasized for achieving the highest true negative accuracy at 99.55%. This characteristic indicates it would contribute most positively to the system for the aforementioned reasons.

Using only vertical data, the models incorporating angle information demonstrate a clear improvement in true positive accuracy compared to GaitBase's 81.60%, achieving 86.85% and 86.81% for GaitBase_In and GaitBase_InN, respectively. This improvement underscores the significance of incorporating angle information into the models, ultimately enhancing their capability to distinguish between new and previously observed individuals.

5. Conclusion

The study effectively integrated angle data into various DL model-free approaches, supported by an innovative but simple gait acquisition system that is expected to be adaptable to varied scenarios through appropriate hyperparameter tuning. The high-quality sequences and angle information enable more accurate identification of multiple subjects and reliable differentiation between known and new individuals. Additionally, the results show that both early and late integration of angle information can lead to improvements, with the optimal strategy depending on the specific model. For instance, models like GaitPart, GaitSet, and GaitGL performed better with late integration, while GaitBase benefited more from early integration. These findings suggest that the choice of integration strategy should be tailored to the model's architecture to achieve the best performance. Notably, including angle information proved beneficial in all models tested on the CASIA-B dataset, with GaitBase_In also showing significant improvements on the CASIA-E dataset. These results highlight the practical advantages of incorporating angle data. The successful implementation of a threshold-based approach for differentiating between new and registered subjects further underscored the practical viability of the proposed system.

In the future, the goal is to test the system as a whole in real-time, integrating the acquisition system with the recognition algorithm through implementing an efficient segmentation algorithm, which remains to be explored and would allow a seamless connection between the two parts. Future research could also focus on further evaluating the contribution of angle information to feature extraction in model-based approaches, where its impact is expected to be more significant due to the structural nature of the data.

This robust gait acquisition and recognition system lays a solid foundation for future enhancements and practical deployment, demonstrating strong potential for real-world applications and significant theoretical and practical advancements.

CRediT authorship contribution statement

Diogo R.M. Bastos: Investigation, Data collection, Code implementation, Formal analysis, Original draft preparation. **João Manuel R.S. Tavares:** Conceptualization, supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This article partially results from the project "Sensitive Industry", co-funded by the European Regional Development Fund (FEDER) through the Operational Programme for Competitiveness and Internationalization (COMPETE 2020) under the PORTUGAL 2020 Partnership Agreement.

Data availability

The data is freely available.

References

Cao, Z., Hidalgo, G., Simon, T., Wei, S.-E., & Sheikh, Y. (2019). OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. arXiv:1812.08008.

Chao, H., Wang, K., He, Y., Zhang, J., & Feng, J. (2021). GaitSet: Cross-view gait recognition through utilizing gait as a deep set. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(7), 3467–3478. http://dx.doi.org/10.1109/TPAMI.2021.3057879.

Cosma, A., Catruna, A., & Radoi, E. (2023). Exploring self-supervised vision transformers for gait recognition in the wild. *Sensors*, *23*(5), http://dx.doi.org/10.3390/s23052680.

Fan, C., Liang, J., Shen, C., Hou, S., Huang, Y., & Yu, S. (2023). OpenGait: Revisiting gait recognition toward better practicality. *Vol. 2023-June*, In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (pp. 9707–9716). http://dx.doi.org/10.1109/CVPR52729.2023.00936, arXiv:2211.06597.

Fan, C., Peng, Y., Cao, C., Liu, X., Hou, S., Chi, J., Huang, Y., Li, Q., & He, Z. (2020). GaitPart: Temporal part-based model for gait recognition. In *2020 IEEE/CVF conference on computer vision and pattern recognition* (pp. 14213–14221). http://dx.doi.org/10.1109/CVPR42600.2020.01423.

Gupta, S., & Chattopadhyay, P. (2020). Exploiting pose dynamics for human recognition from their gait signatures. *Multimedia Tools and Applications*, *80*(28–29), 35903–35921. http://dx.doi.org/10.1007/s11042-020-10071-9.

Gupta, S., & Chattopadhyay, P. (2021). Gait recognition in the presence of co-variate conditions. *Neurocomputing*, *454*, 76–87. http://dx.doi.org/10.1016/j.neucom.2021.04.113.

Harris, E., Khoo, I.-H., & Demircan, E. (2022). A survey of human gait-based artificial intelligence applications. *Frontiers in Robotics and AI*, *8*, http://dx.doi.org/10.3389/frobt.2021.749274.

Hawas, A., El-Khobby, H., Elnaby, M., & El-Samie, F. (2019). Gait identification by convolutional neural networks and optical flow. *Multimedia Tools and Applications*, *78*(18), 25873–25888. http://dx.doi.org/10.1007/s11042-019-7638-9.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. arXiv:1512.03385.

Hermans, A., Beyer, L., & Leibe, B. (2017). In defense of the triplet loss for person re-identification. ArXiv, abs/1703.07737.

Iwashita, Y., Sakano, H., Kurazume, R., & Stoica, A. (2021). Speed invariant gait recognition-The enhanced mutual subspace method. *PLoS ONE*, *16*(8 August), http://dx.doi.org/10.1371/journal.pone.0255927.

Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8.

Khaliluzzaman, M., Uddin, A., Deb, K., & Hasan, M. J. (2023). Person recognition based on deep gait: A survey. *Sensors*, *23*(10), http://dx.doi.org/10.3390/s23104875.

Kidziński, Ł., Yang, B., Rajagopal, A., Delp, S., & Schwartz, M. (2020). Deep neural networks enable quantitative movement analysis using single-camera videos. *Nature Communications*, *11*, 4054. http://dx.doi.org/10.1038/s41467-020-17807-z.

Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C. L., & Dollár, P. (2015). Microsoft COCO: Common objects in context. arXiv:1405.0312.

Lin, B., Zhang, S., Wang, M., Li, L., & Yu, X. (2022). GaitGL: Learning discriminative global-local feature representations for gait recognition. arXiv:2208.01380.

Perez, E., Strub, F., de Vries, H., Dumoulin, V., & Courville, A. (2017). FiLM: Visual reasoning with a general conditioning layer. arXiv:1709.07871.

Ripic, Z., Nienhuis, M., Signorile, J. F., Best, T. M., Jacobs, K. A., & Eltoukhy, M. (2023). A comparison of three-dimensional kinematics between markerless and marker-based motion capture in overground gait. *Journal of Biomechanics*, *159*, Article 111793. http://dx.doi.org/10.1016/j.jbiomech.2023.111793, URL: https://www.sciencedirect.com/science/article/pii/S0021929023003640.

Russel, Shebiah, N., Selvaraj, & Arivazhagan (2021). Gender discrimination, age group classification and carried object recognition from gait energy image using fusion of parallel convolutional neural network. *IET Image Processing*, *15*(1), 239–251. http://dx.doi.org/10.1049/ipr2.12024, arXiv:https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/ipr2.12024.

Salehian, S., Sebastian, P., & Sayuti, A. B. (2019). Framework for pedestrian detection, tracking and re-identification in video surveillance system. In *2019 IEEE international conference on signal and image processing applications* (pp. 192–197). http://dx.doi.org/10.1109/ICSIPA45851.2019.8977728.

Sayeed, M. S., Min, P. P., & Bari, M. A. (2022). Deep learning based gait recognition using convolutional neural network in the COVID-19 pandemic. *Emerging Science Journal*, *6*(5), 1086–1099. http://dx.doi.org/10.28991/ESJ-2022-06-05-012.

Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., & Sun, J. (2018). CrowdHuman: A benchmark for detecting human in a crowd. arXiv:1805.00123.

Song, C., Huang, Y., Wang, W., & Wang, L. (2023). CASIA-E: A large comprehensive dataset for gait recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(3), 2801–2815. http://dx.doi.org/10.1109/TPAMI.2022.3183288.

Takemura, N., Makihara, Y., Muramatsu, D., Echigo, T., & Yagi, Y. (2018). Multi-view large population gait dataset and its performance evaluation for cross-view gait recognition. *IPSJ Transactions on Computer Vision and Applications*, *10*, http://dx.doi.org/10.1186/s41074-018-0039-6.

Upadhyay, J., & Gonsalves, T. (2022). An enhanced gait recognition system based on the features fusion methodology with recurrent neural network (RNN). *Indian Journal of Computer Science and Engineering*, *13*(5), 1483–1496. http://dx.doi.org/10.21817/indjcse/2022/v13i5/221305042.

Wang, Y., Song, C., Huang, Y., Wang, Z., & Wang, L. (2019). Learning view invariant gait features with Two-Stream GAN. *Neurocomputing*, *339*, 245–254. http://dx.doi.org/10.1016/j.neucom.2019.02.025.

Xianda, G., Zheng, Z., Tian, Y., Beibei, L., Junjie, H., Jiankang, D., Guan, H., Jie, Z., & Jiwen, L. (2022). Gait recognition in the wild: A large-scale benchmark and NAS-based baseline. arXiv:2205.02692.

Xu, K., Jiang, X., & Sun, T. (2021). Gait recognition based on local graphical skeleton descriptor with pairwise similarity network. *IEEE Transactions on Multimedia*, *24*, 3265–3275. http://dx.doi.org/10.1109/TMM.2021.3095809.

Yu, S., Tan, D., & Tan, T. (2006). A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. *Vol. 4*, In *18th international conference on pattern recognition* (pp. 441–444). http://dx.doi.org/10.1109/ICPR.2006.67.

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2022). ByteTrack: Multi-object tracking by associating every detection box. In *Proceedings of the European conference on computer vision*.

Zheng, L., Zha, Y., Kong, D., Yang, H., & Zhang, Y. (2022). Multi-branch angle aware spatial temporal graph convolutional neural network for model-based gait recognition. *IET Cyber-Systems and Robotics*, *4*(2), 97–106. http://dx.doi.org/10.1049/csy2.12052.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57–81. http://dx.doi.org/10.1016/j.aiopen.2021.01.001.