From Joints to Pressure: Multimodal Deep Learning Methods for Pathological Gait Recognition

Antonio Mattesco[†], Alex Meggiolaro[‡]

Abstract—**Pathological gait classification is a challenging task with significant implications for clinical diagnosis and rehabilitation. In this work, we address the classification of pathological gaits using a publicly available multimodal dataset composed of 3D skeleton sequences and plantar pressure data. We replicate state-of-the-art approaches from related studies to establish a robust baseline. Building upon this foundation, we explore alternative deep learning architectures, including bidirectional GRUs, Transformer-based models and hybrid models. Additionally we evaluate the impact of different preprocessing techniques and introduce a novel feature: the joint-wise speed computed between consecutive frames. The best results were achieved with the Transformer architecture applied to skeleton data, reaching an accuracy of 0.92 while also reducing the number of parameters. We outline possible factors contributing to the discrepancy between our results and those originally reported in the literature.**

Index Terms—**Abnormal gait recognition, multimodal classification, neural networks, hybrid deep learning.**

I. Introduction

Gait, or the manner in which a person walks, is a fundamental indicator of overall health and neurological function, reflecting the complex coordination of the musculoskeletal and nervous systems. Abnormal gait can significantly affect quality of life, making gait analysis a crucial tool for early diagnosis, rehabilitation monitoring, and clinical decision-making. Early detection of gait anomalies enables timely interventions and improved treatment outcomes.

Sensing technologies for gait analysis are typically divided into wearable and non-wearable systems. Wearable sensors, attached directly to the body, provide real-time, portable monitoring suitable for both clinical and everyday settings, though they may cause discomfort and require careful calibration. Non-wearable systems, such as video cameras, depth sensors, or foot pressure mats, observe movement externally. While often less precise, they are less invasive, easier to deploy, and generally more cost-effective.

This study presents a method for classifying five types of pathological gaits (antalgic, lurching, steppage, stiff-legged, and Trendelenburg) as well as normal gait, using 3D skeleton data from Azure Kinect and foot pressure data from the GW1100 system, by applying deep learning techniques. The study focuses on understanding how different data representations and model architectures influence performance. We explore multiple temporal cropping strategies to determine the impact of sequence segmentation on classification accuracy. Additionally, we evaluate a range of deep learning models, including those incorporating attention mechanisms, to better capture spatial-temporal dependencies inherent in gait data. We also introduce walking speed as a novel input feature, derived from skeleton trajectories, and assess its contribution to classification performance.

The main paper contributions are:

- Exploration of different deep learning architectures for gait classification, including CNNs, RNNs, RNN-Autoencoder and Transformer based architectures.
- Investigation of temporal cropping strategies to assess how different segment lengths influence model accuracy.
- Introduction of walking speed as an additional input feature, computed from skeleton data, and analysis of its impact on classification results.

The structure of this report is as follows: Section II reviews the state of the art in abnormal gait recognition. Section III provides an overview of the proposed methodology, which is further detailed in Section IV (data preprocessing) and Section V (model architectures). Section VI presents the results, and Section VII concludes the report with final remarks.

II. Related Work

In the literature, a wide variety of sensor technologies have been employed to analyze human gait, ranging from wearable sensors to vision-based systems. Wearable sensors, such as inertial measurement units (IMUs) and planar foot pressure sensors, provide direct physical measurements of movement and ground reaction forces. On the other hand, non-contact systems such as depth cameras and optical motion capture systems estimate human motion from visual input, often through the reconstruction of 3D skeletal data.

Wearable sensors are particularly effective for extracting gait parameters such as stride length, velocity, and phase durations with high precision. IMUs, for instance, have been used with classical machine learning techniques such as support vector machines (SVMs) and hidden Markov models (HMMs) to classify different walking patterns [1]. Similarly, foot pressure data collected through force platforms or pressure mats have proven valuable for detecting abnormal gait types, especially in clinical environments [2].

Despite their lower raw accuracy compared to direct sensors, skeleton-based methods obtained via depth cameras are attractive due to their non-invasiveness, lower cost, and ability to cover longer walkways. Although 3D skeletal data are indirectly derived from depth images and therefore subject to

[†] Department of Mathematics, University of Padova, email: {antonio.mattesco}@studenti.unipd.it

[‡] Department of Mathematics, University of Padova, email: {alex.meggiolaro.1}@studenti.unipd.it

estimation errors, they remain widely used in abnormal gait recognition, especially in real-world or outpatient settings.

Several works have leveraged machine learning models to classify pathological gait types using skeletal data. [3] proposed a method that classified normal and various abnormal gaits, including antalgic, stiff-legged, lurching, steppage, and Trendelenburg gaits, with an accuracy of 93.67%, by feeding selected joint trajectories into a GRU-based classifier. Since this approach required manual selection of informative joints, later work proposed automated feature extraction pipelines, such as RNN-based autoencoders [4], which learned latent representations from raw skeleton sequences, later used for classification via recurrent models.

In parallel, other studies focused on foot pressure data, exploring both handcrafted feature extraction and end-to-end learning methods [5]. These works demonstrated that spatial and temporal pressure patterns are particularly informative for distinguishing specific gait disorders, such as those associated with neurological or musculoskeletal impairments.

Recently, multimodal approaches have gained traction for combining complementary information from multiple types of sensors. Fusing data from skeletal and foot pressure sources allows models to leverage both motion dynamics and ground contact patterns. Extracting features separately from both modalities and combining them using recurrent neural network classifiers improves classification performances [6]. In [7] the effect of early versus late fusion strategies is investigated, and it is found that timing and method of integration significantly impacted classification performance.

In this context, the work in [8] explores the integration of joint velocity as an additional feature for pathological gait recognition. Their method, based on spatiotemporal graph convolutional networks and attention mechanisms, showed that including velocity information alongside joint positions significantly improved classification accuracy.

III. Processing Pipeline

Pathological gaits are classified using a Convolutional Neural Network (CNN) based approach for foot pressure data and a Recurrent Neural Network (RNN) and Transformer based approach for skeleton and speed data. Additionally, we experimented with a hybrid model that integrates both data, aiming to leverage complementary information from multiple sources to improve classification performance.

To support this, a structured preprocessing pipeline tailored to each modality and the corresponding modeling is implemented.

a) ***Data Preprocessing:*** Before application in any Machine Learning or Neural Network algorithm, the data must be properly preprocessed. The dataset used in this study is publicly available in [9] and consists of 1,440 examples: for each of the 12 subjects, 20 trials are recorded for each of the five pathological gaits, as well as for the normal gait. Each trial includes time-stamped 3D skeleton joint data along with a corresponding foot pressure image. The limited amount of data is alleviated with data augmentation techniques. Specifically, for skeleton data we performed left-right joint swapping, while for foot pressure images we applied horizontal flipping doubling the total number of samples. Due to the differing nature of these modalities, images and time series, distinct preprocessing techniques are required for each.

- **Skeleton Data**: Each example is a variable-length sequence (118 to 509 time steps) capturing 3D coordinates (x, y, z) of 32 joints (96 features per step). Noisy joints are excluded [3], and time steps with zero time difference are removed. The data are then normalized per example, and a relevant subsequence is cropped for analysis.
- **Speed Data**: Joint speeds are derived from the skeleton sequences, with the same removal of zero-difference time steps. The data are normalized per example and temporally aligned using the same cropped subsequence as the skeleton data.
- **Foot Pressure Data**: Each sequence is associated with a fixed-size grayscale image $(128 \times 48 \times 1)$ representing foot pressure distribution. Images are normalized and centered.

A detailed explanation of the full preprocessing pipeline can be found in Sec. IV.

b) ***Models Training:*** Separate models are trained for each modality: a Convolutional Neural Network (CNN) is used to classify foot pressure images, as CNNs are well-suited for spatial feature extraction and have proven effective in various image classification tasks. For skeleton and speed data, Gated Recurrent Units (GRUs) are employed due to their ability to capture temporal dependencies in sequential data.

In the case of skeleton data, additional experiments are conducted using Transformer-based architectures, which have gained popularity in time series analysis due to their strong ability to model long-range dependencies.

Furthermore, Long Short-Term Memory (LSTM) Autoencoders are explored as an alternative to manual joint selection. Initially, joint subsets proposed in [3] are used to reduce dimensionality and noise. However, inspired by the approach in [4], an LSTM-based Autoencoder is developed to automatically extract high-level, compact representations from the full joint sequence. Autoencoders are specific neural networks trained to reconstruct their input, thereby learning meaningful latent features. When combined with LSTM layers, they are particularly well suited for modeling temporal dynamics and compressing sequential data. The resulting embeddings are then fed into a GRU-based classifier for gait recognition.

Moreover, to combine the different data sources in a unique hybrid architecture, the single models are repurposed as feature extractors after training, by trimming them before their final classification layers. The extracted features from each modality are concatenated and passed to a fully connected net serving as the final classifier for pathological gait recognition. The model output is a probability vector representing the likelihood of belonging to each of the six classes.

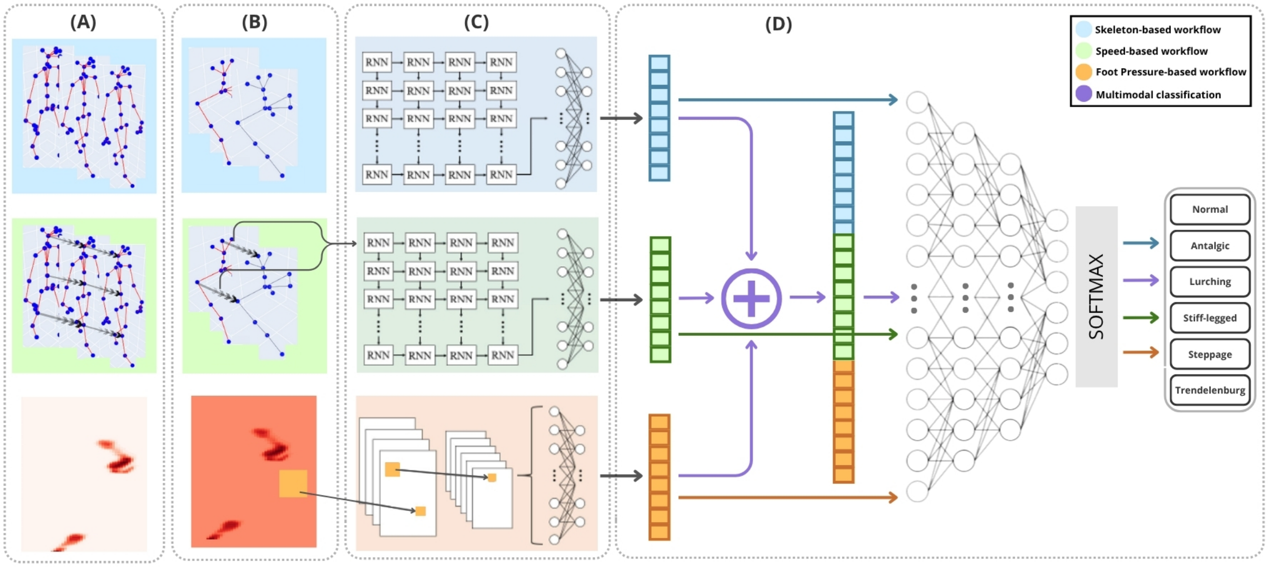

Fig. 1: Overview of the proposed workflow. (A) The input consists of three data modalities: skeleton joints (blue), joints speed (green), and foot pressure (orange). (B) Skeleton and speed data are first filtered by removing time steps with zero time difference, and a subset of joints is discarded. All modalities are then normalized, and foot pressure images are also centered. (C) Each modality is processed through its respective model to extract discriminative features. (D) The models are first trained independently; subsequently, their extracted features are concatenated and passed to a final FC layer for classification.

IV. Signals and Features

Skeleton and foot pressure data are collected using the Azure Kinect sensor and a GW1100 pressure plate. The Kinect provides 3D coordinates of 32 joints, calibrated using an ArUco marker placed on a 4-meter walkway. The pressure data are captured using a high-resolution sensor mat (1080mm×480mm, 6,144 sensors). For full details regarding the acquisition setup and calibration procedure, refer to [7]. The raw input data are organized in folders by subject. Each subject folder contains subfolders for each gait type, which in turn contain 20 trial folders. Each trial folder includes two csv files: one with spatial and temporal skeleton joint data, and one with the corresponding foot pressure image data.

Each gait sequence can be represented as a matrix $\boldsymbol{X} = [x_{ij}]$, where $i = 1, \dots, T_x$ corresponds to the number of recorded frames, and $j = 1, \dots, 96$ represents skeleton joint coordinates (3D coordinates for each of the 32 joints). However, only 14 joints related to the legs and spine are retained, as the others add noise and little informative value. This selection yields a matrix $\boldsymbol{X} \in \mathbb{R}^{T_x \times 42}$.

Each sequence is then normalized independently using a robust percentile-based method to mitigate the effect of outliers. Specifically, for each joint coordinate x, the normalization is computed as:

$$\boldsymbol{x}^{norm} = \frac{\boldsymbol{x} - P_5(\boldsymbol{x})}{P_{95}(\boldsymbol{x}) - P_5(\boldsymbol{x})} \tag{1}$$

where P_5 and P_{95} denote the 5th and 95th percentiles of the coordinate values over the sequence, respectively.

Finally, a cropping strategy is applied to ensure that all sequences have a fixed temporal length K. To achieve this, several methods are implemented:

1) **aggressive center**: Given a sequence $\boldsymbol{X} = [x_{ij}]$ of length T, cropping boundaries are defined by removing noisy parts:

$$t_{\text{start}} = \frac{T}{2} - 20, \quad t_{\text{end}} = T - 10, \tag{2}$$

extracting the subsequence of length $T' = t_{\text{end}} - t_{\text{start}}$:

$$\boldsymbol{X'} = [x_{ij}], \quad i = t_{\text{start}}, \dots, t_{\text{end}}. \tag{3}$$

Then, a fixed-length subsequence of size K is center-cropped within X' by setting

$$t'_{\text{start}} = \frac{T'}{2} - \frac{K}{2}, \tag{4}$$

and finally

$$\boldsymbol{X''} = [x_{ij}], \quad i = t'_{\text{start}}, \dots, t'_{\text{start}} + K. \tag{5}$$

This approach removes noisy frames at the start and end, extracting a centered subsequence of length K.

2) **aggressive random**: Similar to *aggressive center*, but instead of a centered crop, the starting index t'_{start} is randomly selected uniformly in the interval $[0, T' - K]$, thus cropping a random subsequence of length K within the trimmed region.
3) **center**: Same as *aggressive center* but with less aggressive trimming, setting $t_{\text{start}} = 10$ and $t_{\text{end}} = T - 5$, focusing on the more central portion of the original sequence.
4) **random**: Similar to *center* for trimming, but a random subsequence of length K is selected within the trimmed range.
5) **top**: Selects the initial portion of the sequence after skipping the first 10 frames to remove transient noise, then takes the next K frames.
6) **bottom**: Selects the last portion of the sequence, removing the final 25 frames and cropping the last K frames remaining.
7) **40perc**: Discards the first 40% of the sequence and crops the next K frames from the remaining part. This is designed to approximately capture the moments when subjects are on the pressure plate.
8) **sliding window**: Extracts all consecutive, non-overlapping subsequences of length K by sliding a window over the entire sequence after applying the *40perc* crop, increasing data availability. This acts as a data augmentation technique, since each subsequence is treated independently.

Speed data are not directly available. To compute them, for each joint in a skeleton sequence with corresponding time vector, the spatial displacement between consecutive frames is calculated as the Euclidean distance between the joint's 3D coordinates at time t and $t + 1$. The temporal difference is computed from the corresponding time values. The speed for each joint at time t is then obtained by dividing the spatial displacement by the temporal difference. This results in a sequence where each time step is represented by 14 speed features, one per joint. The preprocessing steps, including normalization and cropping, are applied to speed data as described previously for skeleton data.

The foot pressure data are represented as single-channel images, where each pixel corresponds to the pressure value measured by sensors in the pressure plate. Preprocessing includes normalization by scaling raw pressure values to the $[0, 1]$ range via division by 255. To reduce variability due to inconsistent foot positioning, each foot pressure image is centered using its barycenter. This is done by computing the weighted average of pixel coordinates:

$$\bar{x} = \frac{\sum_{i,j} j \cdot I_{ij}}{\sum_{i,j} I_{ij}}, \quad \bar{y} = \frac{\sum_{i,j} i \cdot I_{ij}}{\sum_{i,j} I_{ij}}, \tag{6}$$

where I_{ij} is the pixel intensity at row i and column j. The image is then shifted so that $(\bar{x}, \bar{y})$ aligns with the image center. This normalization ensures that the model learns from pressure distribution patterns rather than foot placement variability.

To evaluate each architecture, a Leave-One-Subject-Out Cross Validation (LOSO CV) strategy is implemented. In this approach, all samples from one subject are excluded and used as the validation set, while data from the remaining subjects serve as the training set. This process is repeated as many times as the number of subjects, so that each subject is used once as the validation set. This method ensures evaluation on completely unseen subjects, avoiding any leakage of subject-specific information into the training phase. Including data from the same subject in both training and validation could result in overestimated performance due to individual-specific patterns, which would not be available during deployment or inference. Additionally, one subject is entirely excluded from the procedure and reserved as an independent test set to assess the model's generalization capability.

V. Learning Framework

This section details the learning strategy and algorithm developed to solve the problem of pathological gait recognition. Several learning architectures tailored to different data modalities are proposed. For skeleton data, various models are explored, including GRU-based architectures, a Transformer-based model, and a GRU-based model applied to features extracted by an Autoencoder. Due to lower discriminative power, speed data are processed using a standalone GRU model. In contrast, foot pressure data are modeled using convolutional neural networks (CNNs).

A. Proposed RNN Architecture

Given the temporally structured nature of both skeleton and speed data, an RNN-based architecture is proposed to model the sequential dynamics of human gait. The network is trained using the preprocessed input sequences, with the goal of learning discriminative temporal patterns across different gait types.

The core of the model consists of a deep stack of Gated Recurrent Units (GRUs), designed to capture the temporal dynamics of skeleton joint sequences. The network takes as input a sequence of preprocessed joint coordinates and outputs a class prediction over six gait categories.

Formally, let $x_t \in \mathbb{R}^d$ denote the input feature vector at time step t, and $h_t \in \mathbb{R}^h$ the hidden state. A GRU cell updates its hidden state as follows:

$$h_t = (1 - z_t) \odot \tilde{h}_t + z_t \odot h_{t-1}, \tag{7}$$

where the update gate z_t, reset gate r_t, and current memory content $\tilde{h}_t$ are computed as:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{8}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{9}$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h) \tag{10}$$

Here, σ denotes the sigmoid activation function, $\odot$ is the element-wise product.

The architecture begins with a dense projection layer with ReLU activation applied to the input sequence. This is followed by four stacked GRU layers:

- The first two GRU layers contain 256 units each and return full sequences.
- The next two GRU layers contain 128 units each. The final GRU layer outputs a single vector.

A dense layer of 100 units, without activation function, convert the features to the desired size. For speed data the numbers are halved. All GRU layers can optionally be bidirectional, enabling the model to learn both past and future dependencies. The resulting features vector is passed through a series of fully connected layers with 128, 64, and 32 units and ReLU activation functions. Before the dense layers, dropout (0.5) and batch normalization are applied. A final fully connected layer with 6 units and softmax produces class probabilities.

B. Proposed Transformer Architecture

A Transformer-based model (TFM) is proposed for sequential input data. The proposed architecture builds upon the standard Transformer model, adapting it to handle time series data and to perform classification instead of language modeling. The architecture incorporates several key components described below.

a) Input and Positional Encoding: The input tensor passes through a learnable positional encoding layer to inject temporal order information into the model, allowing it to capture sequential patterns. Each position is associated with a trainable vector added to the input.

b) Transformer Encoder Blocks: Each encoder block consists of a Multi-Head Self-Attention mechanism followed by a feed-forward network (FFN), both equipped with residual connections and dropout. Layer Normalization is applied before the FFN, which consists of two 1D convolutions with dropout in between.

c) Global Pooling and Classification: After six stacked encoder blocks, a Global Average Pooling layer aggregates the sequence dimension, followed by a multilayer perceptron consisting of three dense layers with 128, 64, and 32 units respectively, each followed by dropout (0.2) and GELU activation function. L2 regularization is also applied. The final classification is performed by a softmax output layer with 6 units, one for each gait class.

C. Proposed Autoencoder Architecture

In order to learn compact and informative representations of skeleton-based gait sequences, a Recurrent Neural Network (RNN) Autoencoder architecture is proposed. The model is designed as an undercomplete autoencoder, where the latent code dimension p is smaller than the input dimension. This encourages the network to capture only the most salient and discriminative features of the data, effectively acting as a learned dimensionality reduction. In this scenario, all joints are kept.

a) Encoder: The encoder consists of four stacked recurrent layers based on LSTM units. Given an input sequence $\mathbf{x}_{1:T} \in \mathbb{R}^{T\times 96}$, each of the first three layers maintains the original feature dimensionality with 96 hidden units. The final layer projects the output to a compact latent representation of size $p < 96$, acting as the bottleneck of the autoencoder.

b) Decoder: The decoder is a single LSTM layer that reconstructs the temporal sequence from the latent code. It takes as input the encoded vector and outputs a reconstructed sequence $\hat{\mathbf{x}}_{1:T} \in \mathbb{R}^{T\times 96}$, matching the original input shape.

c) From Autoencoder to Classifier: The autoencoder is trained to minimize the reconstruction error between the input sequence $\mathbf{x}_{1:T}$ and the output $\hat{\mathbf{x}}_{1:T}$. The loss function used is the mean squared error:

$$\mathcal{L}_{\text{AE}} = \sum_{t=1}^{T} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2^2 \tag{11}$$

This encourages the autoencoder to learn representations that preserve the essential structure and dynamics of gait while compressing the sequence into a compact latent vector. Once trained, only the encoder part is retained and used to extract compressed features from the input sequences. These features are then passed to a separate GRU-based classifier (described in Sec.V-A), which is trained to predict the gait category. This design allows the encoder to act as a data-driven feature extractor, avoiding arbitrary decisions regarding joint selection.

D. Proposed CNN Architecture

Average foot pressure data resemble single-channel image data. For this reason, a Convolutional Neural Network is employed, which is well known for its strong performance in image processing and classification tasks, to extract meaningful features from the foot pressure inputs. The proposed architecture is composed by a sequence of convolutional blocks, followed by a dense stage for feature projection and final classification. The convolutional component consists of four distinct blocks: the first three blocks each include two convolutional layers with 3×3 kernels, ReLU activations, and padding, while the fourth block comprises a single convolutional layer. The four convolutional blocks increase progressively the number of feature maps from 16 to 128. Each block is followed by a 2×2 MaxPooling operation, progressively reducing the spatial resolution of the feature maps and enabling a more abstract representation of the input while decreasing computational complexity. After the final pooling operation, the extracted features are flattened and passed through a sequence of three fully connected layers with 512, 256, and 128 units respectively, each using ReLU activation. A dropout layer with a rate of 0.5 is applied to mitigate overfitting, followed by batch normalization. Finally, the output is processed through a series of additional dense layers with 128, 64, and 32 units, regularized via L2 penalty. The network concludes with a dense layer of 6 units with

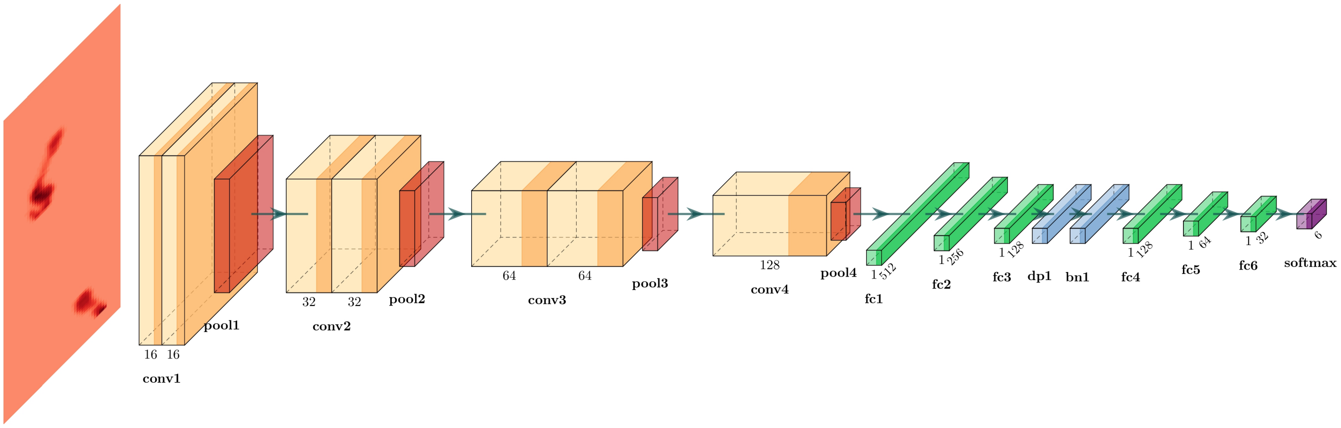

Fig. 2: Proposed Convolutional Neural Network architecture

softmax activation, producing a probability distribution over the target classes.

E. Fusion of Skeleton, Speed and Foot Pressure Data

While the advantages of multimodal fusion have been outlined in the related works, this section presents the architectural details of the proposed hybrid classification framework. The focus is placed on the integration of skeleton motion, foot pressure, and gait speed data through a feature level fusion strategy designed to enhance classification performance. Meaningful representations from each modality are extracted using convolutional neural networks for foot pressure data and recurrent neural networks for both skeleton and speed time series. These networks are truncated at the dense layer before dropout, and the resulting feature vectors are concatenated to form a single, unified representation. Specifically, let $f_{fp} = (f_{fp}^{(1)}, \ldots, f_{fp}^{(N_{fp})})$, $f_{sp} = (f_{sp}^{(1)}, \ldots, f_{sp}^{(N_{sp})})$ and $f_{sk} = (f_{sk}^{(1)}, \ldots, f_{sk}^{(N_{sk})})$ denote the features extracted from foot pressure, speed, and skeleton data, respectively, where N_{fp}, N_{sp}, N_{sk} represent the number of features derived from each modality. The final fused feature vector is given by:

$$f_{fus} = \text{concatenate}(f_{fp}, f_{sp}, f_{sk}) \tag{12}$$

Skeleton data captures the spatiotemporal structure of gait, which is fundamental for recognizing and classifying different gait patterns. Nonetheless, some pathological gaits may present similar joint trajectories, making it challenging to distinguish between them using only vision-based features. Foot pressure data complements this by providing spatial information on weight distribution, which is often affected in pathological conditions. Gait speed data introduces an additional temporal dynamic that can further enhance classification performance.

The concatenated features were input to a DNN-based classifier consisting of four fully connected layers with 128, 64, 32, and 6 units, respectively. Prior to entering the dense layers, dropout (0.5) and batch normalization were applied. All dense layers used the ReLU activation function, except for the final one, which employed a softmax activation to output class probabilities.

F. Training Details

Classification models were evaluated using a LOSO-CV strategy. In each iteration, the data from one subject were held out as the validation set, while the remaining data were used for training. This process was repeated 11 times, once for each subject (one subject is kept out as test), and the average validation accuracy, along with other statistical metrics, was computed.

Model training was guided by the cross-entropy loss function, as expressed by the following formula:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N} \log(\hat{p}_{i,y_i}) \tag{13}$$

where $\hat{p}_{i,y_i}$ is the predicted probability for the right class y_i, $y_i \in \{0, ..., 5\}$ and N is the number of samples in a batch.

Early stopping and L2 regularization to the fully connected layers were applied to mitigate overfitting. In most cases, training was halted before reaching 200 epochs.

Speed and skeleton models were trained using the Adam optimizer with a learning rate of 10^{-4}. For foot pressure and fusion models, the learning rate decreases to 5×10^{-5}. All models use a batch size of 30.

To enhance the classification performance of the multimodal hybrid model, a multi-step training strategy was adopted [6]. First, the individual CNN and RNN based models were trained separately using only foot pressure, skeleton, and speed data, respectively. This step enabled each encoder to specialize in extracting meaningful features from its respective modality. Finally, the hybrid model was initialized with the pre-trained encoders, and end-to-end training was performed to fine tune all weights jointly. This separate training approach was intended to enhance the feature extraction capabilities of the CNN and RNN encoders by allowing each to specialize on its respective modality.

VI. Results

This section presents the results of classifying pathological gaits using the proposed architectures. Model performance was evaluated using LOSO-CV to ensure consistency and generalizability. For each subject, training and validation losses, as well as accuracy metrics, were recorded. In addition to accuracy, we report precision, sensitivity and specificity, to provide a comprehensive assessment of each model's performance. To assess generalization performance, the results of the best model on the held-out test set are also reported.

A. Cropping Strategies Evaluation

The first step was to identify the most effective cropping strategy for input sequences. To this end, a GRU-based model was trained on skeleton data using various cropping methods, and the results are summarized in Tab. 1. They suggest that the initial time steps may contain noise and offer limited discriminative information, as evidenced by the poor performance of *top crop*. Cropping strategies that remove a larger portion from the beginning of the sequence, such as *aggressive center* or *aggressive random*, generally outperform their non-aggressive counterparts.

Among the evaluated cropping strategies, *sliding window* achieved the highest accuracy (0.8932). Its superior performance is likely due to the fact that it functions not only as a cropping method but also as a form of data augmentation. Based on these results, it was selected as the default cropping strategy for all subsequent experiments, using the GRU model as reference and assuming that its effectiveness generalizes across different model architectures.

Crop Type	Time Steps	Accuracy
sliding window	50	0.8932
40perc	100	0.8879
aggressive random	50	0.8780
bottom	100	0.8712
aggressive center	50	0.8621
center	100	0.8129
random	100	0.7538
top	100	0.4515

TABLE 1: Average validation accuracy with respect to crop strategies and number of time steps considered for the GRU model applied to skeleton data.

Modality	Model	Accuracy	Precision	Sensitivity	Specificity
Skeleton	GRU	0.8932	0.8944	0.8924	0.9785
	BiGRU	0.9114	0.9133	0.9114	0.9823
	TFM	**0.9174**	**0.9200**	**0.9174**	**0.9835**
	AE+GRU	0.8811	0.8814	0.8811	0.9762
Foot	CNN	0.5758	0.5753	0.5758	0.9152
Speed	GRU	0.5023	0.5108	0.5023	0.9005

TABLE 2: Average validation performance metrics of models across data modalities

B. Classification Performances of the Single Models

Next, various models were evaluated using single data modalities. As shown in Tab. 2, foot pressure and speed data proved to be less discriminative, achieving accuracies of 0.57 and 0.50, respectively. In contrast, the skeleton standard GRU achieved solid performance, which was further improved by its bidirectional variant, reaching an accuracy of 0.9114. The transformer-based model outperformed all others, achieving the highest accuracy of 0.9174. This superior performance is likely due to the transformer's ability to capture complex spatial-temporal patterns; its self-attention mechanism allows the model to weigh the importance of each time step globally, making it particularly well-suited for structured motion data. Conversely, the model combining an autoencoder-based feature extractor with a GRU classifier underperformed relative to expectations, achieving an accuracy of 0.88. In this context, several code sizes in the range of 50 to 80 were tested, with 60 ultimately selected as it provided the best accuracy.

C. Classification Performances of the Fusion Models

The results of the multimodal fusion models are reported in Tab. 3. All possible combinations of the data modalities were tested. Notably, none of the fused models outperformed the best single-modality model, the Transformer trained solely on skeleton data. This indicates that, in the current setup, skeleton data remains the most informative source.

When considering the GRU architecture, the addition of foot pressure data led to a modest performance improvement, suggesting a degree of complementary information. In contrast, speed data alone provided little to no benefit and, in some cases, introduced noise. Interestingly, when speed data was combined with foot pressure data, the model showed a more appreciable improvement, indicating that even modalities with low individual discriminative power can contribute meaningfully when used together.

Modality	Accuracy	Precision	Sensitivity	Specificity	Δ Acc (%)
SK+SP+FT	0.9030	0.9036	0.9030	0.9806	+1.10
TFM SK+FT	0.8621	0.8600	0.8583	0.9789	-6.02
SK+FT	**0.9053**	**0.9076**	**0.9053**	**0.9811**	+1.35
SK+SP	0.8917	0.8918	0.8917	0.9783	-0.17
SP+FT	0.6197	0.6167	0.6197	0.9239	**+7.62**

TABLE 3: Average validation performance metrics of fusion models. The accuracy variation is computed with respect to the best single model among the ones combined.

However, combining foot pressure data with the Transformer model resulted in a performance drop. This degradation is likely due to the imbalance in feature representation within the hybrid architecture: the Transformer branch extracts a relatively small number of features (42), compared to GRU (100), whereas the CNN processing foot pressure data outputs a significantly larger feature set (128). As a result, the classifier may disproportionately rely on the more dominant foot pressure features, which are less informative, thereby hindering overall performance.

Modality	Model	Training Time (h)	Time per Epoch (s)	Number of Parameters	Size (MB)	GPU Memory Usage (GB)
Skeleton	GRU	1.10	2.78	910K	3.47	0.21
	BiGRU	2.03	5.03	2.5M	9.47	0.54
	TFM	1.01	3.05	677K	2.58	0.73
	AE+GRU	6.46	13.33(AE)+2.05(GRU)	1.1M	3.90	ROOM
Foot	CNN	0.47	1.37	1.9M	7.29	0.73
Speed	GRU	0.82	2.86	220K	0.88	0.20
SK+SP+FT	Fusion	3.55	6.67^{*}	3M	11.58	ROOM
SK+SP	Fusion	2.54	5.68^{*}	1.1M	4.33	0.38
SK+FT	Fusion	2.40	6.07^{*}	2.8M	10.72	0.81
TFM SK+FT	Fusion	2.27	6.04^{*}	2.5M	9.98	ROOM
SP+FT	Fusion	2.04	7.87^{*}	2.1M	8.15	0.66

TABLE 4: Models comparison based on computational metrics. *correspond to LOSO of the fusion model only (excluding separate training of SK, SP, and FT). ROOM indicates LOSO run split due to memory limits.

These findings diverge from prior works where multimodal fusion led to significant improvements. A key distinction lies in the architectural choices: for instance, in [6], foot pressure data was processed using a large pretrained DenseNet, capable of extracting more meaningful representations. In this study, a smaller, non-pretrained model was used for foot pressure data, likely constraining its effectiveness.

In summary, while multimodal fusion can be beneficial under certain conditions, particularly when combining weak but complementary modalities, the best results in this setting were achieved using skeleton data alone with a Transformer model. The effectiveness of fusion appears to depend critically on both the informativeness of each modality and the preservation of strong architectural components during integration.

D. Computational Cost of the Models

The computational requirements of the evaluated models are summarized in Tab. 4. As expected, fusion architectures are the most demanding in terms of training time and GPU memory usage, due to the complexity of processing multiple input modalities. Among the single-modality skeleton models, the Transformer achieves the best trade-off between performance and efficiency. It has the lowest number of parameters (677K) and the fastest training time, outperforming both the standard GRU (910K parameters) and the BiGRU (2.5M parameters). However, it also exhibits the highest GPU memory usage during training, likely due to the overhead introduced by the self-attention mechanism. The Autoencoder-based model shows significantly higher training time due to the two-stage training process involving both the autoencoder and the GRU classifier. The models based on foot pressure and speed data are not particularly demanding in terms of parameters or memory usage, but their limited performance reduces their overall utility.

All experiments were conducted using an NVIDIA Tesla T4 GPU, freely available on Google Colab.

E. Additional Results of the Transformer Model

All the results presented in this section refer to the Transformer model trained on skeleton data, which achieved the best performance in terms of both evaluation metrics and computational efficiency.

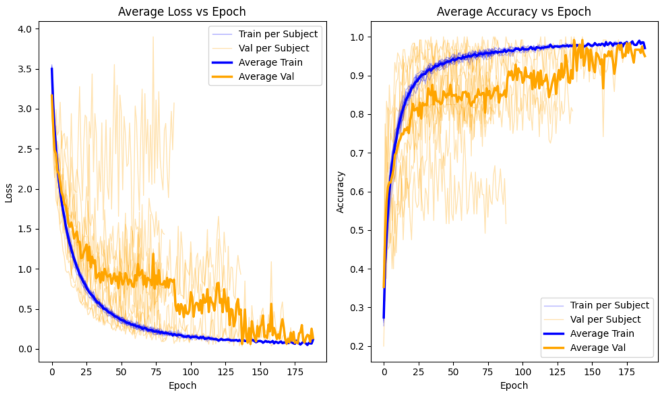

Fig. 3: Training and validation loss (left) and accuracy (right) across subjects for the Transformer model.

Fig. 3 shows the training and validation losses and accuracies across epochs. Bold lines represent the average performance, while thinner lines indicate the individual results for each subject in the LOSO-CV. Overall, the learning process appears effective; however, some subjects (9 and 10), when used as validation sets, fail to reach the performance levels observed for other subjects. This outcome further highlights the importance of the implemented training procedure and aligns with the findings in [8], which emphasize the lack of diverse datasets. The gap between training and validation accuracy indicates that the model is learning effectively, but also suggests potential overfitting, despite the application of multiple mitigation strategies such as early stopping, dropout, and regularization. The validation loss exhibits significant fluctuations, indicating instability and difficulty in consistently capturing generalized features. The high and inconsistent validation loss, combined with the noticeable gap between training and validation metrics, suggests that the model struggles to generalize effectively. This issue is likely influenced by a small number of problematic subjects. The average accuracy across validation and test sets of 0.9174 and 0.9136, respectively, demonstrates strong overall model performance. The test accuracy indicates effective generalization to unseen data. Furthermore, the similarity between validation and test accuracies suggests that the model is neither significantly overfitting nor underfitting.

Gait	Accuracy	Precision	Sensitivity	Specificity
Normal	0.9705	0.9171	0.9045	0.9836
Antalgic	0.9576	0.8254	0.9455	0.9600
Stiff-legged	0.9932	0.9607	1.0000	0.9918
Lurching	0.9848	0.9587	0.9500	0.9918
Steppage	0.9591	0.8990	0.8500	0.9809
Trendelenburg	0.9697	0.9592	0.8545	0.9927

TABLE 5: Average validation performance metrics of Transformer model per gait

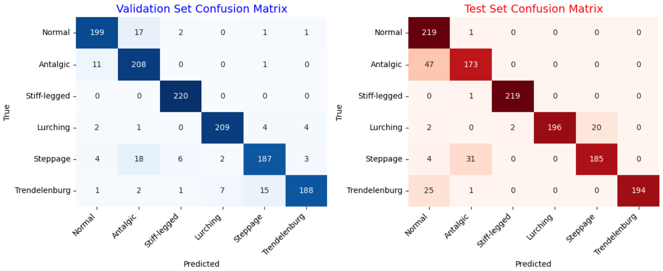

Fig. 4: LOSO confusion matrices of the Transformer model on validation (left) and test (right) sets

Finally, Tab. 5 presents the average validation performance metrics of the transformer model for each gait class. Notably, the Normal class achieves high specificity (0.9836) indicates a low false positive rate for this class, confirming the model's reliability in distinguishing normal gait from pathological ones. The best performance is observed in the Stiff-legged class, likely because the semicircular leg dragging pattern results in more distinctive joint coordinate trajectories, making it easier for the model to capture relevant features.

VII. Concluding Remarks

A. *Summary, observations and future works*

This study explored various deep learning architectures for the classification of pathological gait patterns, with a focus on the temporal dynamics of skeletal joint trajectories. The initial experiments highlighted the importance of selecting meaningful temporal segments within the input sequences. Cropping strategies that removed the initial noisy time steps, particularly the sliding window approach, yielded better results.

Among the tested architectures, Transformer-based model consistently outperformed others, achieving the best trade-off between performance (accuracy of 0.9174) and number of parameters (677K).

Fusion models that integrated skeleton, speed, and foot pressure data did not improve performance over the Transfomer model for skeleton data only. This result contrasts with prior studies and is attributed to the limited discriminative power of the additional modalities in our dataset. Specifically, our CNN model for foot pressure was lightweight and non-pretrained, which may have restricted its ability to extract meaningful features. In comparison, other works rely on large pretrained networks (e.g., DenseNet201) for image processing, achieving higher accuracies (0.68). Nonetheless, combining foot and speed data alone led to modest improvements over individual performance, indicating potential complementarity. This suggests that multimodal fusion remains a promising direction, provided that each modality is represented by strong and well-modeled features.

Future work could explore hybrid approaches that combine Transformer-based architectures for skeleton data with larger pre-trained models for foot pressure data (e.g., DenseNet201). Additionally, a sliding window ensemble strategy could be employed to extract multiple subsequences per example, enabling a voting mechanism that may improve prediction robustness.

B. *What we have learned*

Throughout this project, we gained valuable insights into designing hybrid architectures that integrate different data types and neural paradigms. We learned the critical importance of preprocessing and of choosing appropriate validation strategies; in particular, the LOSO-CV setup proved essential for realistic performance evaluation. A standard train/validation/test split would have failed to capture inter-subject generalization challenges.

This has been the most demanding project we have undertaken, both in terms of technical complexity and effort required. Despite initial challenges and setbacks, we are satisfied with the outcomes. Some model and processing variants yielded clear improvements over our baseline, while others did not; however, through iterative experimentation and analysis, we were able to steadily improve our results and draw meaningful conclusions.

References

[1] R. Caldas, M. Mundt, W. Potthast, F. B. de Lima Neto, and B. Markert, "A systematic review of gait analysis methods based on inertial sensors and adaptive algorithms," *Gait & posture*, vol. 57, pp. 204–210, 2017.

[2] A. S. Alharthi, A. J. Casson, and K. B. Ozanyan, "Gait spatiotemporal signal analysis for parkinson's disease detection and severity rating," *IEEE Sensors Journal*, vol. 21, no. 2, pp. 1838–1848, 2020.

[3] K. Jun, Y. Lee, S. Lee, D.-W. Lee, and M. S. Kim, "Pathological gait classification using kinect v2 and gated recurrent neural networks," *Ieee Access*, vol. 8, pp. 139881–139891, 2020.

[4] K. Jun, D.-W. Lee, K. Lee, S. Lee, and M. S. Kim, "Feature extraction using an rnn autoencoder for skeleton-based abnormal gait recognition," *IEEE Access*, vol. 8, pp. 19196–19207, 2020.

[5] M. Wang, S. Yong, C. He, H. Chen, S. Zhang, C. Peng, and X. Wang, "Research on abnormal gait recognition algorithms for stroke patients based on array pressure sensing system," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 1560–1563, IEEE, 2019.

[6] K. Jun, S. Lee, D.-W. Lee, and M. S. Kim, "Deep learning-based multimodal abnormal gait classification using a 3d skeleton and plantar foot pressure," *IEEE Access*, vol. 9, pp. 161576–161589, 2021.

[7] M. T. Naseem, H. Seo, N.-H. Kim, and C.-S. Lee, "Pathological gait classification using early and late fusion of foot pressure and skeleton data," *Applied Sciences*, vol. 14, no. 2, p. 558, 2024.

[8] J. Kim, H. Seo, M. T. Naseem, and C.-S. Lee, "Pathological-gait recognition using spatiotemporal graph convolutional networks and attention model," *Sensors*, vol. 22, no. 13, p. 4863, 2022.

[9] K. Jun, S. Lee, D.-W. Lee, and M. S. Kim, "Azure kinect 3d skeleton and foot pressure data for pathological gaits," 2021.