Compact Hybrid Neural Network for Pathological Gait Detection Using Skeleton and Pressure Data

Barbiero Lorenzo[†], Zane Francesco[‡]

***Abstract*—In the context of automatic human gait recognition, using data from non-wearable devices is crucial for real-world applicability. In this study, we design a hybrid learning framework to efficiently analyze skeletal time series collected with a depth camera, along with pressure measurements collected during walking. After developing a comprehensive data cleanup pipeline to ensure high-quality data for our learning model, we test several small and lightweight hybrid neural network architectures. We select the best model that can quickly and accurately detect pathological traits in the walking motion of subjects across six different classes. The resulting model is a hybrid CNN-based architecture with just 750k parameters (2.75 MB). It achieves a global validation accuracy of 86.9% and a class-specific accuracy greater than 93%. This efficient and accurate model could enable the widespread implementation of gait recognition algorithms in everyday devices with minimal impact on resources and performance, potentially improving health diagnostics and monitoring.**

***Index Terms*—Abnormal gait classification, multi modal sensors, neural networks, hybrid machine learning.**

I. Introduction

Human gait refers to the manner or pattern of walking, encompassing the movements and mechanics of the legs and feet during locomotion. It is important for health as it affects mobility, balance, and coordination. Abnormal, or pathological, gait can indicate underlying health issues such as neurological disorders, musculoskeletal problems, or injuries. Proper gait assessment can aid in diagnosing conditions, planning rehabilitation, and preventing falls, thereby improving overall quality of life and functional independence.

Given the significant impact gait has on health, the ability to accurately assess and recognize gait patterns is crucial. This is where automatic gait recognition comes into play, offering advanced methods to analyze gait more efficiently and precisely.

In the context of automatic gait recognition, methods can vary significantly. Some approaches use predetermined, complex equations to compute gait parameters explicitly, while others rely on machine learning frameworks to make predictions. Another key difference is the type of sensor used for measurements: wearable versus non-wearable sensors. While wearable technology can potentially provide more accurate and diverse data, they come with the main drawback that users must attach sensors or markers to their bodies, which can cause discomfort, and the process can be cumbersome, invasive, and expensive, potentially difficult to implement in real-world scenarios. The trend in recent years has been to focus on non-wearable devices so that these classifications, given their importance for health, can be made more frequently and in the simplest and most economical way possible.

In this study, leveraging data collected by non wearable sensors, we propose a novel method for classifying five pathological gaits (antalgic, lurching, steppage, stiff-legged, and Trendelenburg) and normal gait using 3D skeleton data from Microsoft Azure Kinect and foot pressure data from the GW1100 Pressure Plate.

To perform accurate predictions we will use a neural network architecture that aims at taking advantage of the information contained in both types of data simultaneously in order to improve the quality of the analysis.

As previously mentioned, our contribution to the state of the art emphasizes real-world applicability. By developing models from scratch, we achieved high performance with significantly lower computing and memory requirements than previous approaches. This results in a slim and lightweight architecture with fewer than 750k parameters and a total weight of 2.75 MB.

The main contributions of this paper are:

- we developed an architecture designed for real-world, everyday applications that achieves high performance while keeping the number of parameters very limited.
- we confirmed that fusing skeleton and foot pressure data positively impacts pathological gait classification by capturing both spatio-temporal motion and weight distribution.

This report is structured as follows: in Section II we present the related state of the art works. In Section III we outline our method, which is explained in more detail in Section IV for the data processing and in Section V for the machine learning models. Results are discussed in Section VI while conclusions are provided in Section VII.

II. Related Work

The main source of inspiration for this report came from the exceptional work done by Kooksung, Jun et Al., which also provided the dataset used in the research [1].

In the paper *Abnormal Gait Recognition Using 3D Joint information of Multiple Kinects System and RNN-LSTM* [2], published in 2019, the data acquisition process is quite complex and cumbersome. The fact that six different Kinect sensors are needed and the resulting raw data needs interpolation

[†] Department of Physics, University of Padova, email: {lorenzo.barbiero.1}@studenti.unipd.it

[‡] Department of Physics, University of Padova, email: {francesco.zane.1}@studenti.unipd.it

makes the endeavor expensive, complex and time consuming. The main takeaway from this paper is the importance of proper joint selection for an accurate prediction. In this initial stage the best architecture is a LSTM feature extractor followed by a feed-forward neural network classifier. The final dataset was comprised of more than 50000 50-frames sequences and led to an accuracy of roughly 97%.

The paper *Feature Extraction Using an RNN Autoencoder for Skeleton-Based Abnormal Gait Recognition* [3], published in January 2020, aims to classify a different set of eighth anomalous gaits which are left/right specific and don't directly stem from diagnosed gaits, examples are weights or padding soles on one certain foot. The architecture is largely similar, but a recurrent autoencoder for feature extraction followed by a classifier was also explored. This approach yielded satisfactory results, achieving up to 95% validation accuracy on approximately 2000 useful data sequences.

On Paper *Pathological Gait Classification Using Kinect v2 and Gated Recurrent Neural Networks* [4], published in July 2020, further research is developed on the basis of Paper [3]. A more comprehensive metrics set is evaluated, including class specific accuracy, sensitivity, specificity and precision.

The most relevant breakthrough happened in the paper *Deep Learning-Based Multimodal Abnormal Gait Classification Using a 3D Skeleton and Plantar Foot Pressure* [5], published in 2021 with a new, simpler, data acquisition method which could make the measurement process more applicable to real world scenarios. Technology advancements made it possible to use only one Kinect for the whole measurement, with the main drawback being the fact that the useful walking distance for measurements was shorter (from 6 m to 4 m). There is also the introduction of pressure data as a secondary data source that can allow for multimodal data analysis. The best resulting model is multi-modal with a GRU-RNN for the skeleton part and the pre-trained DenseNet201 for the pressure data which achieved a validation accuracy of 93.4%.

III. Processing Pipeline

In this experiment a combination of two distinct data types have been collected, which from now on will be referred to as *Skeleton Data* and *Pressure Data.*
In this study, we classified six different types of gait using both skeleton and pressure data. The dataset consisted of 12 healthy individuals who simulated one normal gait and five abnormal gaits. The skeleton data was collected using a Kinect sensor, while the pressure data was gathered from a pressure plate. In the initial stages of our architecture, these two types of data were processed separately to extract meaningful information from each. They were then combined only in the final stage.

- *Phase 0*: Preprocessing on the data samples was carried out, where they were normalized and cleaned of any potential noise.
- *Phase 1*: In this phase, we developed an architecture consisting of a sequentially linked feature extractor and classifier designed to classify different types of gait using only Skeleton data.
- *Phase 2*: Similarly to Phase 1, we designed an additional architecture that includes a sequentially linked feature extractor and classifier to classify different types of gait, this time utilizing only Pressure data.
- *Phase 3*: In the final phase, we developed an architecture capable of integrating information from both Skeleton and Pressure data. Specifically, it includes the two previously mentioned feature extractors and a classifier. The input vector for the classifier is derived from concatenating the output features obtained from both the skeleton and pressure feature extractors.

IV. Signals and Features

The data acquisition was performed by *Kooksung, Jun et Al.* and the collected data can be found at [1].

During the process 12 healthy male subjects were asked to mimic six different types of gaits: the involved individuals were previously instructed by and fully understood the key features of each specific gait in order to have a walking motion as close as possible to the real pathological one.

1) *Normal gait*
2) *Antalgic gait:*
   - Biological cause: Pain in foot, ankle, knee, or pelvis
   - Replicated by: Keeping the weight off the injured leg
3) *Stiff-legged gait:*
   - Biological cause: Stiffness of knee or hip
   - Replicated by: Avoiding bending the knee
4) *Lurching gait:*
   - Biological cause: Weakness or paralysis of the gluteus maximus muscle
   - Replicated by: Extending the torso backwards to compensate hip mobility
5) *Steppage gait:*
   - Biological cause: Weakness or paralysis of the anterior tibialis muscle
   - Replicated by: Raising the knee higher than normal
6) *Trendelemburg gait:*
   - Biological cause: Weakness or paralysis of the gluteus medius muscle
   - Replicated by: Changing hip level mid gait

These pathological type of gaits were accurately chosen to allow for repeatability and reproducibility of results even for healthy individuals. Unlike a sensory problem or Parkinson's related gait, which involves random irregularities in the walking pattern, all gaits studied in this research stem from well studied joint or muscular complications and are thus more accurately reproducible in a laboratory setting.

Skeleton data tracks the movement of the subject through time; it has been collected using a Microsoft Azure Kinect, which is a depth camera that automatically detects 32 joints of the human body in 3D space and returns a time series as an output.

Pressure data was collected using a GHiWell GW1100 pressure plate. It returns a 128x48 pixel image containing the average foot pressure over the walking action.

Subjects were instructed to walk a 4 meter walkway with the pressure plate in the middle so that both the skeletal time series and the pressure acquisitions could be made at the same time.

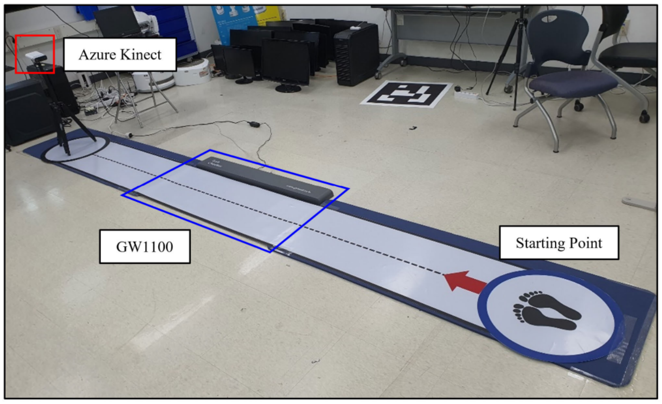

Fig. 1: System setup for multimodal data collection [5]

For each subject 20 samples of each gait were collected, so the total amount of data samples is $num\ subjects \times num\ gaits \times num\ trials = 12 \times 6 \times 20 = 1440\ samples$.

After importing the data the raw structures of the datasets were:

- *Skeleton:* arrays of shape (frames = variable, joints = 32, axes = 3). The average sequence length was around 220 frames.
- *Pressure:* arrays of shape (height = 128, width = 48, channels = 1).

Before feeding the data into our learning pipeline some important preprocessing had to be performed, especially on the skeletal data.

Preprocessing skeleton data: the first challenge to tackle was to have clean ad high quality informative samples while wasting as little data as possible so a three step data cleanup and selection method was implemented:

- *Remove the initial noisy data:* inspecting the data, we realized that the beginning of the measurement did not coincide with the beginning of the walking action by the subject; oftentimes, the first several frames would contain no movement at all or unrelated actions. For this reason, almost the first half of each measurement had to be removed as noise.
- *Data normalization:* we performed [0,1] data normalization to ensure that each feature is on a common scale, which allows the neural network to process the data more effectively. This normalization step not only enhances the stability and speed of the training process by reducing the risk of exploding or vanishing gradients, but it also ensures that each input feature contributes equally to the learning process.
- *Joint selection:* as highlighted by [2], there is the possibility that the motion of some joints is not correlated to the walking pattern and could, on the contrary, lead to unnecessary noise in learning. For this reason we implemented a function which could filter out selected body joints and the latter learning architectures have been trained both using the full skeleton and using only a restricted joints set (legs and spine, with the exclusion of the head, shoulders and both arms).

To increase the data variety, two augmentation techniques were implemented:

- *Random flip:* each time the sample is presented in training there is a random chances to have it flipped along the x direction, which corresponds to inverting the left and right side of the body. The flip parameter was set to 0.5.
- *Random crop:* with data cleaned of the noisy portion each time the sample is presented in training a different window of length 50 is chosen, in this way the full data can be used through different epochs.

Preprocessing pressure data: in this case the preprocessing was slightly less involved and featured less steps:

- *Normalization:* data was normalized in the $[0, 1]$ range along the channel.
- *Centering:* the barycenter of each sample was computed and used to shift the image accordingly, in order to have it at the center of the image.

The augmentation only featured the *random flip* along the x coordinate similarly to what has been described above. It is important to highlight that the flipping of the two data sample was always synchronized.

After the pre-processing had been performed the structures of the datasets were:

- *Skeleton:* arrays of shape (frames = 50, joints, axes = 3) or (frames = 50, 3 $\times$ joints) depending on the architecture
- *Pressure:* arrays of shape (height = 128, width = 48, channels = 1)

Train/test split:

Finally, we outline the train/test split of the data and the thought process behind it. Given that individual tendencies and biases would almost surely impact the training algorithm a *leave one out* approach was chosen, to avoid such unwanted correlations as much as possible. For this reason subject number 12 (*Subject 12* from now on) was left out from the beginning and used as the Test set while the other 11 subjects were part of the training set and were used in a *leave one out* cross validation.

V. Learning Framework

In this section, we provide a detailed description of the encoder architectures used for extracting features from Skeleton and Pressure data. We will then explain how we integrated the information from these two data types to enhance the model's classification performance.

A. Skeleton encoder

Regarding skeleton data, we developed various approaches to extract meaningful information:

- *RNN*
- *CNN*
- *CNN inception*

The data used in training featured either the full 32 joints available or just 12 selected joints (legs and spine).

RNN

Skeleton data are time series, so our initial approach was to implement an RNN encoder. This encoder consists of four layers with $256, 256, 128$, and 128 hidden units, respectively. The first three layers return sequences as outputs, while the last layer returns only the final element of the sequence. To prevent the vanishing gradient problem while keeping the number of parameters limited, we used GRU layers with $\tanh$ as activation function.

CNN

Our second approach was to develop a CNN encoder. In this setting, the input time series were treated as images with dimensions (height, width, channels) corresponding to (number of frames, number of joints, space dimension). The architecture is composed of 16 layers, organized into four groups. Each group consists of the same sequence of four layers:

1) Conv2D (filters, kernel, padding = same)
2) BatchNormalization
3) Activation function = ReLu
4) MaxPooling2D

We modified the Conv2D layers in our neural network by changing the kernel size and the number of filters for each layer. Specifically, we used kernel sizes of $(3, 3)$, $(3, 3)$, $(5, 5)$, and $(5, 5)$, and set the number of filters to $16, 32, 64$, and 128, respectively. Throughout these layers, we consistently applied the ReLu activation function and used padding set to 'same'. For the MaxPooling2D layers, we used a pool size of $(2, 2)$, with pool strides and padding set to 2 and 'same', respectively.

CNN Inception

We developed an additional architecture inspired by the Inception model [6]. This architecture aims to enhance the expressive capabilities of the CNN approach, enabling the model to discover relationships between not neighboring frames and joints. The main idea was to implement a layer similar to the Inception model, but instead of altering the size of the Conv2D kernels, we varied the stride values.

In detail, the architecture of our Inception layer shown in figure 2 is structured as follows:

- Input data is from the previous layer is passed through three separate (1, 1) Convolutional layers, each with 16 fixed filters. These layers aim to extract the most meaningful information from the input, essential for keeping a small number of parameters.
- The outputs from the previous layers are then passed into three separate 2D Convolutional layers, each with a fixed number of 32 filters and the same kernel size but varying strides $1, 2$ and 3. This approach allows each layer to capture relationships among frames and joints at different time and spatial distances.
- The resulting tensors from the previous layers are concatenated into a single output tensor along the depth axis.

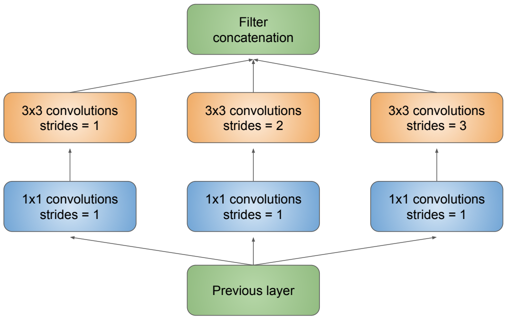
Filter concatenation
3x3 convolutions strides = 1
3x3 convolutions strides = 2
3x3 convolutions strides = 3
1x1 convolutions strides = 1
1x1 convolutions strides = 1
1x1 convolutions strides = 1
Previous layer

Fig. 2: Modified Inception layer

The CNN Inception architecture alternates between Inception layers and a custom layer referred to as Conv2DNoMax. The Conv2DNoMax layer consists of three sequential components:

1) Conv2D (filters, kernel = $(3, 3)$, padding = same)
2) BatchNormalization
3) Activation function = ReLu

The only parameter not fixed within the Conv2DNoMax layer is the number of filters in the Conv2D layer.

The final architecture of the CNN inception was developed as follows:

1) Conv2DNoMax (filters = 16)
2) Inception Layer
3) Inception Layer
4) Conv2DNoMax (filters = 32)

Regardless of the specific encoder architecture, we extracted skeleton-based features $\mathbf{f_{sk}}$ using the following equation:

$$\mathbf{f_{sk}} = \mathbf{W_{sk}} \cdot \text{flatten}(E_{sk}(\mathbf{sk})) + \mathbf{b_{sk}}. \tag{1}$$

Where $\mathbf{sk}$ are skeleton data vectors.

The final dimension of the vector $\mathbf{f_{sk}}$ is fixed and denoted by N_{sk}.

B. Pressure encoder

To classify the pressure data, we focused on an approach using a CNN encoder. The architecture is composed of 16 layers, organized into four groups. Each group consists of the same sequence of four layers.

We customized the Conv2D layers in our neural network by varying both the kernel size and number of filters in each layer. Specifically, we employed kernel sizes of $(3, 3)$, $(3, 3)$, $(3, 3)$, and $(5, 5)$, with number of filters set to $16, 32, 64$, and 128, respectively. Throughout these layers, the ReLU activation function was consistently applied, and we used 'valid' padding. For the MaxPooling2D layers, we utilized a pool size of $(2, 2)$, with both pool strides and padding set to 2 and 'same', respectively.

Similarly to skeleton data, the pressure-based features $\mathbf{f_p}$ were extracted using the following equation:

$$\mathbf{f_p} = \mathbf{W_p} \cdot \text{flatten}(E_p(\mathbf{p})) + \mathbf{b_p}. \tag{2}$$

Where $\mathbf{p}$ are pressure data vectors.

The final dimension of the vector $\mathbf{f_p}$ is fixed and denoted by N_p.

C. Fusion model

In this paper, we developed a classification model that integrates multimodal fusion of Skeleton and Pressure data. Optimal encoder architectures were selected to extract key features from foot pressure and skeletal information, respectively. These features were then fused at the feature level, as illustrated in figure 3.

The combined skeleton and pressure features are integrated into a unified vector through the formula:

$$\begin{aligned} \mathbf{f_{sk+p}} &= \text{concatenate}(\mathbf{f_{sk}}, \mathbf{f_p}) \quad (3) \\ &= (f_{sk}^1, \ldots, f_{sk}^{N_{sk}}, f_p^1, \ldots, f_p^{N_p}) \quad (4) \end{aligned}$$

where N_{sk} and N_p represent the numbers of extracted features from the pressure-based and Skeleton-based encoding layers, respectively.

D. NN classifier

We employed the same classifier architecture for $\mathbf{f_s}$, $\mathbf{f_p}$ and $\mathbf{f_{sk+p}}$.

It consists of a 4-layers feed-forward neural network with 128, 64, 32, and 6 units, respectively. Prior to feeding the features into the fully connected layers, dropout(0.5) and batch normalization were applied. In the first three layers, the activation function used was ReLu. However, in the final layer, we opted for the softmax function:

$$\mathbf{y} = \text{softmax}(\text{CLS}_{\text{NN}}(\mathbf{f})) \tag{5}$$

where $\mathbf{f}$ could be $\mathbf{f_p}$, $\mathbf{f_p}$ and $\mathbf{f_{sk+p}}$ depending on the case, and $\mathbf{y} = (y_1, ..., y_6)$ is the output vector representing the six gait patterns. We applied a cross-entropy loss function to calculate the loss L_{CE} and applied l_2 regularization to avoid overfitting as follows:

$$L_{CE}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{6} y_i \log(\hat{y}_i) + \frac{\lambda}{2}\|\mathbf{W}\|^2 \tag{6}$$

where $\mathbf{W}$ are the trainable weights and λ is the regularization parameter.

In the pressure-based classification model, we extracted features by feeding the foot pressure data into CNN encoding layers. The resulting features, $\mathbf{f_p}$, were then input into NN-based classification layers. Similarly, for the skeleton-based classification model, the features $\mathbf{f_{sk}}$ extracted from the best skeletal gait data encoder were fed into the NN-based classification layers.

In the multimodal classification model, we fed the Pressure and Skeleton data into their respective optimal encoders. The resulting integrated multimodal features, $\mathbf{f_{sk+p}}$, were then input into the NN-based classification layers.

E. Fusion model training

To train the entire multimodal classification architecture, which comprises the best Skeleton and Pressure encoders, we explored three different approaches, named:

- 0-step
- 1-step
- 3-step

0-step: in this approach we first train the skeleton and pressure classifiers separately. We then use these trained encoder models to construct the multimodal classifier, keeping the encoder weights frozen and training only the NN-based classifier.

1-step: with this strategy we construct the multimodal classifier using the optimal skeleton and pressure encoder architectures, but without any prior training. We train the entire model in a single iteration.

3-step: this hybrid approach combines elements of the first two. Initially, we train the skeleton and pressure classifiers separately. Subsequently, using the weights from the trained encoder models, we construct the multimodal classifier. From this starting point, we proceed to train the entire architecture without freezing any weights.

F. Training specification

We evaluated both the single-modal and multimodal classification models using leave-one-subject-out cross-validation. Each evaluation involved using the data of one subject as validation data while using the rest for training. This procedure was repeated 11 times, and we computed the average validation accuracy along with other classification metrics.

To optimize training, we applied early stopping to achieve the best validation accuracy, typically concluding training before reaching 200 epochs. All neural network-based models were implemented using TensorFlow, utilizing the Adam optimizer [7] with a learning rate set to 0.0001 and a batch size of 30.

VI. Results

In this section, we evaluate the performance of different architectures for gait classification. We classified gait patterns using single-modal classification models built from foot pressure or skeleton data to identify the most effective architectures. Once the best performing models were identified, the relative encoder parts were used as feature extractors to develop a multi modal hybrid model. Finally, we evaluate various training methods for the joint architecture by comparing the outputs of the 0-step, 1-step and 3-step training processes.

The main evaluation metric used in choosing the best model was the average global validation accuracy achieved on the 11 Subjects leave one out cross validation. However it is also useful to consider four more class-specific metrics:

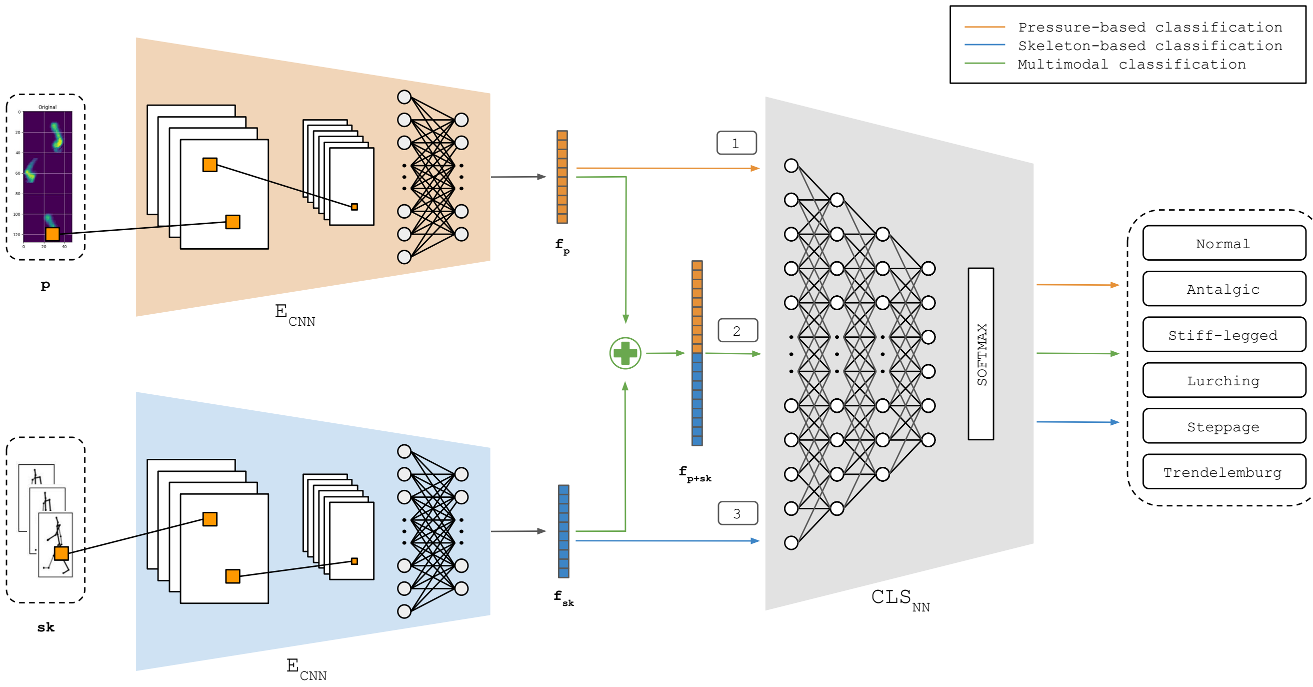

Fig. 3: In both single-modal and multimodal approaches, Skeleton and Pressure data are fed to their respective encoders. In the single-modal setting, the extracted feature vectors $\mathbf{f_p}$ and $\mathbf{f_{sk}}$ are directly passed to the CLS_{NN} classifier. In the multimodal approach, the two feature vectors are concatenated into a single vector, named $\mathbf{f_{p+sk}}$. In the 3-step training method, the single-modal classification processes (1) and (3) are conducted first, and the trained variables are saved. Next, the trained variables from the two E_{CNN} encoders are loaded, and the multimodal classification process (2) is performed.

- Accuracy $= \dfrac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$
- Sensitivity $= \dfrac{\text{TP}}{\text{TP} + \text{FN}}$
- Specificity $= \dfrac{\text{TN}}{\text{FP} + \text{TN}}$
- Precision $= \dfrac{\text{TP}}{\text{TP} + \text{FP}}$

An additional significant element to keep into consideration is the model complexity in terms of number of parameters.

A. *Single mode classification*

The initial step in building an effective multi-modal classifier involved identifying he optimal single-modal classifier architectures for Skeleton and Pressure data separately. The details of the neural networks employed are discussed in SectionV; here, we focus on the results.

Table 1 shows that for the Skeleton data the best performance is achieved by the CNN architecture with only 12 joints that reached a validation accuracy of 84.8%. It is noticeable the increase in performance if joint selection is performed, however the gain changes depending on the architecture type. It is also significant to point out how the two better performing architectures also have the lowest complexity.

Table 2 presents the performance of the CNN-based classifier for Pressure data. In the complexity-performance trade-off, we selected an encoding size of 100 as the optimal value. With this configuration, the architecture achieves a validation accuracy of 43.7%.

TABLE 1: Single mode classification - Skeleton

Model	Num. joints	Parameters	Accuracy
CNN	32	500 k	81.5%
CNN	12	400 k	**84.8%**
RNN	32	975 k	70.0%
RNN	12	925 k	71.6%
CNN In[1]	32	750 k	83.4%
CNN In	12	350 k	83.8%

In general, we observed that the accuracy for the Skeleton data is significantly higher compared to the Pressure data, which is 41.1% lower. This is, however, already been observed in the literature [5] and could be consequence of the smaller amount of available data that makes the bias-complexity trade-off much more challenging.

For the best single modal classifier we also present in Table 3 and in Table 4 the advanced metrics.

[1] The acronym *CNN In* refers to the architecture described in Section V, which is named CNN Inception.

TABLE 2: Single mode classification - Pressure

Model	Enc. size	Parameters	Accuracy
CNN	10	250 k	39.8%
CNN	50	285 k	42.5%
CNN	100	330 k	**43.7**%
CNN	200	420 k	43.3%
CNN	500	690 k	**43.7**%
CNN	1000	1140 k	43.6%
CNN	1500	1590 k	43.5%

TABLE 3: Best model - Skeleton

Gait Type	Acc(%)	Sens(%)	Spec(%)	Prec(%)
Antalgic	91.2	76.4	94.2	72.4
Lurching	97.2	91.4	98.4	91.8
Normal	94.9	89.5	96.0	81.7
Steppage	94.2	78.6	97.4	85.6
Stiff-legged	99.1	96.4	99.6	98.1
Trendelenburg	94.2	80.5	97.0	84.3

Across both tables we can notice that Antalgic and Steppage gaits scored the worst performances in general.

Regarding skeleton-based classification results, the accuracies for each gaits are higher than 91%, while the value of sensitivity is lower than 80% only for Antalgic and Steppage gaits.

In Table 4, pressure-based classification results show that the overall gaits specific accuracy is higher than 80%. However, sensitivity and precision, with the exception of the Normal gait, reached a maximum of 45.9% and 43.5%, respectively.

B. Multi-mode classification

Using the best models from the single-modal classification we build the joint model. We also experimented using different types of training process, named: 0-Step, 1-Step and 3-Step.

Unsurprisingly, the 1-Step model is less accurate. Training the entire model in one iteration, especially given the significant disparity between the accuracies achieved by the skeleton-based and pressure-based classifiers, could lead to local minima and inaccurate results. 0-Step, while faster in training with respect to 3-Step, does not lead to a significant enough performance gain (0.2%) with respect to skeleton based classifier in order to justify the more complex model. The 3-Step process, on the other hand, offers a meaningful accuracy improvement of 1.4% compared to the best single-modal-based classifier model. The advanced metrics obtained using the 3-step approach are shown in Table 6.

The general trend of the metrics shown in Table 6 is similar to what has been observed for the Skeleton 3. Accuracy is always above 93% and the other metrics denote a general, but not uniform, increase in performance. For example, Antalgic Gait improved at the expense of Steppage Gait.

TABLE 4: Best model - Pressure

Gait Type	Acc(%)	Sens(%)	Spec(%)	Prec(%)
Antalgic	80.5	39.5	88.6	41.0
Lurching	81.1	38.2	89.7	42.6
Normal	88.2	75.0	90.8	62.0
Steppage	76.5	27.7	86.3	28.8
Stiff-legged	80.3	45.9	87.2	41.7
Trendelenburg	81.4	37.7	90.2	43.5

TABLE 5: Multi-mode Classification

Model	val Loss	val. Accuracy
Skeleton-based classifier	0.68	85.5%
Pressure-based classifier	2.27	43.8%
Full model 0-Step	0.56	85.7%
Full model 1-Step	0.74	79.2%
Full model 3-Step	0.50	**86.9**%

Observing the confusion matrix (Figure 4-c) we can gain some more insight on the behavior of the model. The biggest misclassifications observed are:

- Trendelenburg Gait is misclassified as Antalgic Gait 14% of the times
- Steppage Gait is misclassified as Antalgic, Lurching or Normal Gait 5.5%, 5.5% and 10% of the times, respectively

From a more health oriented point of view we can notice that:

- A Normal, healthy gait is misclassified as pathological 4.5% of the times
- A pathological gait is misclassified as Normal 3.9% of the times, 2.0% of which due to the Steppage gait and 1.9% due to all the other Gaits

Finally our best model, the multimodal classificator with the 3-step approach, has been trained on *Subjects[1-11]* and tested on *Subject12*. It reached an exceptional test accuracy of 96.7% (Figure 4-d). However, the statistical significance of the test is not very solid due to the limited sample size and would need more thorough evaluation.

VII. Concluding Remarks

A. Summary, observations and what's next

In this project we proposed a new approach to an existing line of studies, with the core strength being in the nimbleness of the developed models, looking at real world applicability.

By already having advanced cameras and depth sensors on last generation smartphones, along with the increased focus on health applications for technology it is feasible to think that, in a near future, automatic gait recognition could be implemented even in a local smartphone environment. With light models taking up little resources in the storage, SOC (System On Chip) and battery departments, this framework could alert the user if anomalies are detected leading to sooner diagnosis and intervention by medical professionals.

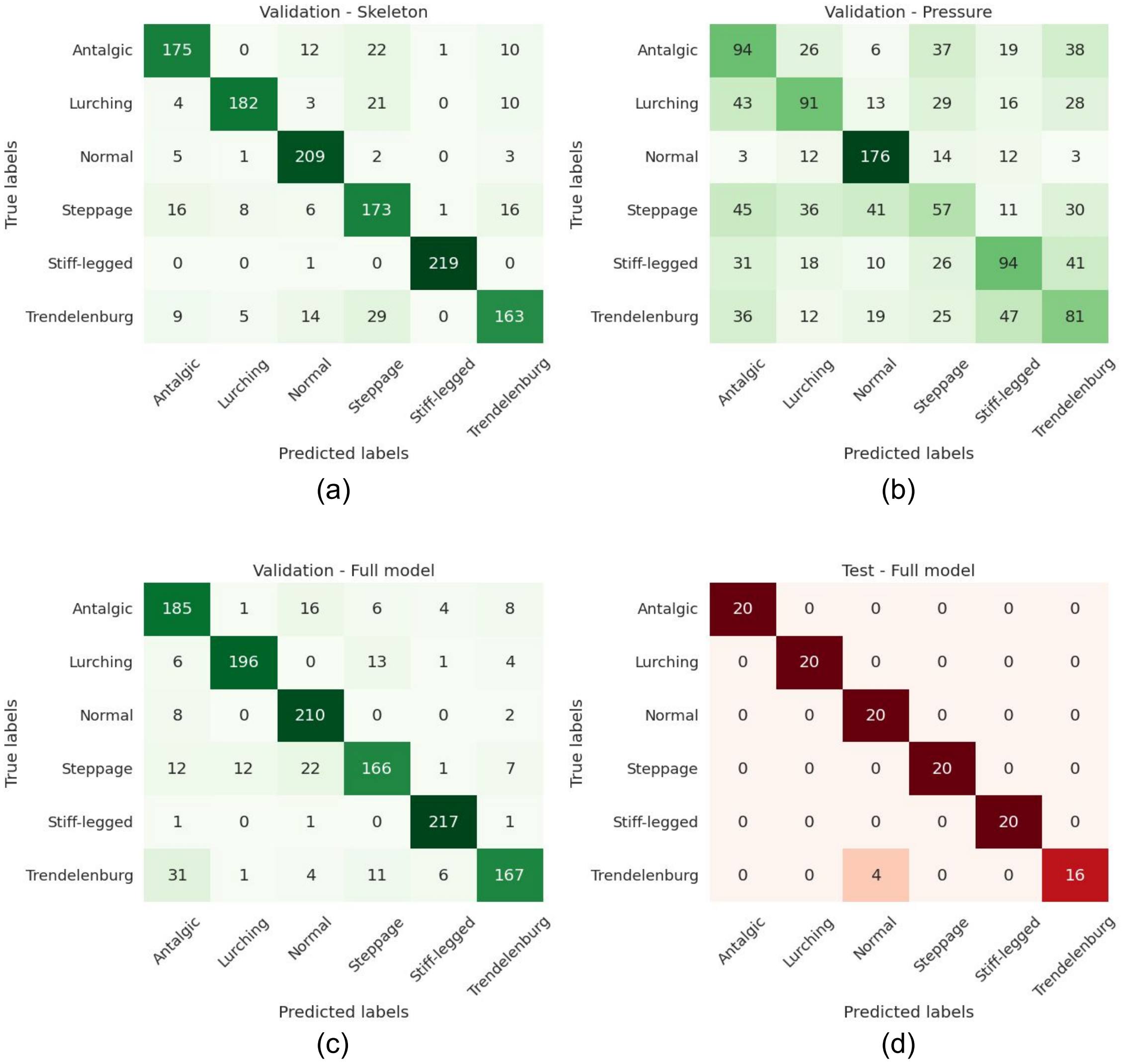

Fig. 4: cross validation and test set confusion matrices: (**a**) Skeleton based classifier validation; (**b**) Pressure based classifier validation; (**c**) 3-Step Full Model validation; (**d**) 3-Step Full Model test

TABLE 6: Full model 3-step

Gait Type	**Acc(%)**	**Sens(%)**	**Spec(%)**	**Prec(%)**
Antalgic	93.0	84.1	94.7	76.1
Lurching	97.1	89.1	98.7	93.3
Normal	96.0	95.5	96.1	83.0
Steppage	93.6	75.5	97.3	84.7
Stiff-legged	98.9	98.6	98.9	94.8
Trendelenburg	94.3	75.9	98.0	88.4

Following with this reasoning, however, we find pressure data much more difficult to acquire outside a laboratory/clinical setting. Fortunately, skeleton data showed good performance even by itself so it can still be a good option for everyday use, while improvements returned by the hybrid model can be reserved to more specialized clinical exams.

B. What we have learned

We learned how to implement architecture that leverages more than one data type at the same time and also understood and dealt with the increased challenges of multiclass classification. We did not manage to replicate the success of the previous research group regarding the RNN architectures, despite being the ones on which we worked and experimented the most. Similarly in preliminary testing on the Pressure dataset we did not manage to retrieve useful results by using big pre-trained models as DenseNet201 since it quickly devolved into over fitting. Still, this was our most advanced machine learning challenge to date and, while it has required a lot of work, time and effort, we are very satisfied by the hands-on experience it provided us with.

References

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

[2] D.-W. Lee, K. Jun, S. Lee, J.-K. Ko, and M. S. Kim, "Abnormal gait recognition using 3d joint information of multiple kinects system and rnn-lstm," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 542–545, 2019.

[3] 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.

[4] 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.

[5] 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.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.