IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 26, 2018, accepted December 9, 2018, date of publication December 14, 2018,
date of current version January 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2886899

An End-to-End Multi-Task and Fusion CNN for Inertial-Based Gait Recognition

RUBÉN DELGADO-ESCAÑO[1], FRANCISCO M. CASTRO[1], JULIÁN RAMOS CÓZAR[1], MANUEL J. MARÍN-JIMÉNEZ[2], AND NICOLÁS GUIL[1]

[1]Department of Computer Architecture, University of Málaga, 29071 Málaga, Spain
[2]Department of Computer Science and Numerical Analysis, University of Córdoba, 14071 Córdoba, Spain

Corresponding author: Francisco M. Castro (fcastro@uma.es)

This work has been funded by project TIC-1692 (Junta de Andalucía), TIN2016-80920R (Spanish Ministry of Science and Technology) and a research initiation Grant (no. #75) from the University of Malaga (Campus de Excelencia Internacional Andalucía Tech). The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

ABSTRACT People identification using gait information (i.e., the way a person walks) obtained from inertial sensors is a robust approach that can be used in multiple situations where vision-based systems are not applicable. Typically, previous methods use hand-crafted features or deep learning approaches with pre-processed features as input. In contrast, we present a new deep learning-based end-to-end approach that employs raw inertial data as input. By this way, our approach is able to automatically learn the best representations without any constraint introduced by the pre-processed features. Moreover, we study the fusion of information from multiple inertial sensors and multi-task learning from multiple labels per sample. Our proposal is experimentally validated on the challenging dataset OU-ISIR, which is the largest available dataset for gait recognition using inertial information. After conducting an extensive set of experiments to obtain the best hyper-parameters, our approach is able to achieve state-of-the-art results. Specifically, we improve both the identification accuracy (from 83.8% to 94.8%) and the authentication equal-error-rate (from 5.6 to 1.1). Our experimental results suggest that: 1) the use of hand-crafted features is not necessary for this task as deep learning approaches using raw data achieve better results; 2) the fusion of information from multiple sensors allows to improve the results; and, 3) multi-task learning is able to produce a single model that obtains similar or even better results in multiple tasks than the corresponding models trained for a single task.

INDEX TERMS Gait, inertial, CNN, fusion, multi-task.

I. INTRODUCTION

Gait is an unequivocal biometric pattern of human locomotion since each subject has its own biological characteristics, making viable the unambiguous identification of people by their way of walking. Gait analysis can be traced back to 60's, when it was used to study walking patterns from healthy people [1], as well as for the early diagnosis of neurological disorders such as cerebral palsy [2], Parkinson's disease [3] or Rett syndrome [4]. Moreover, this subject of study does not only include topics from the area of medicine, but it has been also explored by other research fields, as security [5].

Thus, gait recognition can be used as a biometric pattern in security applications. Biometric security is defined as a mechanism used to authenticate subjects based on the verification of physical characteristics of the subjects. Unlike the iris, face, fingerprint, palm veins or other biometric identifiers, the gait pattern can be collected in a non-invasive manner. This way, subjects do not have to actively collaborate with the system, what allows gait recognition can be used in complex environments where subjects have to wear special clothes (*e.g.* NBC –nuclear, biological, chemical– suits) or where other biometric patterns cannot be used due to specific limitations (position of the cameras, privacy laws, etc.). Moreover, gait patterns are reliable since they are difficult to duplicate due to their dynamic nature. On the other hand, gait recognition is challenging because differences between walking styles from different people can be very subtle. It can also be affected by transient factors such as fatigue, illness, emotions, etc. In addition, external factors such as clothing, shoes or carrying heavy loads, influence gait [6]. Despite that, many approaches have demonstrated that gait can be used as a biometric pattern for recognition [7]–[9]. Gait patterns are mainly applied to two different tasks: identification [9], [10], where gait is used to obtain the *identity* of a known subject,

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

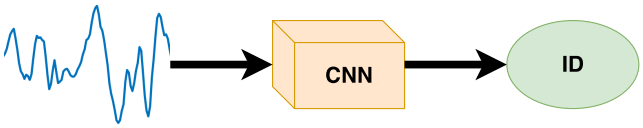

FIGURE 1. Overview. Sketch of the process followed in our approach. The raw inertial information obtained from a sensor is passed through the CNN model to obtain the identity of a subject.

and authentication [11]–[13], where gait is used to validate the *identity* of a known subject.

Commonly, the gait recognition problem has been studied from a computer vision point of view as it is not intrusive for the subject [14]. However, in the last decade, gait recognition using inertial sensors has become an active and exploited topic thanks to the cheapening of Micro Electro Mechanical Systems (MEMS) and their integration into smartphones [15] or smartwatches [16]. This wearable sensor-based approach relies on inertial sensors (*e.g.* accelerometer sensors, gyro sensors, force sensors) placed in different parts of the human body (*e.g.* waist, pockets, shoes, hands) to record gait information.

As a result, wearable inertial sensors have been employed in a variety of research topics related to inertial analysis such as driving analysis [17], fall detection [18], rehabilitation and therapy for patients [19] and surveillance and monitoring of users daily life [20].

In this paper, we propose an end-to-end gait recognition system based on Convolutional Neural Networks (CNN) that uses inertial sensor data as input. Fig. 1 shows a sketch of the process followed in our approach. In addition, two extensions on this CNN-based model are proposed and studied. The first one includes an early fusion scheme that uses information obtained from several sensors. The second one adopts a multi-task scheme to produce multiple outputs from a single input. Specifically, the tasks considered by our model are *identification*, *gender* recognition and *age* estimation. However, in case more labelled data were available, new kinds of tasks could be added. Finally, our proposal is tested on the OU-ISIR Gait Database [21], since it is the largest gait dataset including inertial information. According to the results, our multi-task and fusion approach establishes a new state-of-the-art, evidencing that the use of multiple inputs and multiple outputs benefits the learning process.

Thus, the main contributions of this paper are: *(i)* an end-to-end approach that uses raw inertial data as input; *(ii)* an early fusion scheme to take advantage of information from multiple sensors; *(iii)* a multi-task learning to improve the learning process and to produce multiple outputs from a single input; and, *(iv)* state-of-the-art results on the challenging dataset OU-ISIR Gait Database [21].

The rest of the paper is organized as follows. We start by reviewing related work in Sec. II. Then, Sec. III explains the proposed CNN architectures, and the fusion and multi-task techniques. Sec. IV contains the experiments and results. Finally, we present the conclusions in Sec. V.

II. RELATED WORK

The study of the gait using information from inertial sensors attached to the subject is widely applied to many different fields like human activity recognition [13], [22], fall detection [23], Parkinson's diagnosis [24] or monitoring patients with Parkinson's disease [25], [26]. Another application which is gaining importance during the last years is people identification using their way of walking, *i.e.* gait recognition. In general, most of the gait recognition approaches are based on computer vision [7], [27]–[30], but there are also previous works which are based on inertial sensors [12], [31]. In those examples, the inertial sensor is attached to the subject in a specific position and orientation. Thus, the data collected by the sensor has the same coordinate system. Note that the sensor position is an important aspect to be taken into account as motion dynamics can vary depending on sensor location. Some typical positions are hips [32], legs [33], chest, ankles, lower back and wrists, or combinations of the previous positions [34]. More realistic locations, such as in a bag [35] or in a pocket [20], have also been investigated. Other approaches like [36] focus on the authentication problem using sensors integrated on smartphones so the position is not controlled. Connor and Ross [37] review a wide range of approaches applied to the gait recognition problem using different kind of inputs.

In the field of gait recognition, many different approaches have appeared during the last years. Dynamic Time Warping (DTW) has been used as a distance measure in [38]–[40]. In these works, gait sequences are initially partitioned into gait cycles and compared, using DTW, with some previously selected reference cycles for each class. A similar approach is presented in [41], where, instead of using DTW as metric to compare the cycles, a Hidden Markov Models (HMM) is applied. Similarly, in [42], a cyclic rotation metric (CRM) is employed instead of DTW. Classification Trees are used by Watanabe [43] as classifier for gait recognition. They are employed to process inertial data extracted from the mobile phone of the subjects while they are walking. Choi *et al.* [44] compare different gait signature metrics to represent the gait information. Finally, these signatures are classified with a k-Nearest Neighbors algorithm. Other techniques apply radial basis function (RBF) networks to locally approximate the accelerations and angular velocities [12] to identify subjects.

Another important difference between approaches lays on the type of input data, which can be organized and pre-processed in many different ways. Kwapisz *et al.* [45] employ a combination of time domain features such as average, time between peaks or binned distributions. Khandelwal and Wickström [46] proposed a new methodology based on utilizing the fundamental spectral relationship between the movement of different body parts during gait. More complex features are also used, like Time Frequency Representation [34], which is a way to describe a signal simultaneously in a frequency and time space. Higher-Order Statistics [10] are extensions of second-order measures to

higher orders, useful to non-Gaussian's real-life signals as gait signals.

With the advent of deep learning and Convolutional Neural Networks (CNNs), instead of developing features manually, the features are automatically obtained by the network during the training process using raw signals as input. In the case of inertial information, there are two main approaches. On the one hand, the models are fed with raw information coming from the inertial sensors [13]. On the other hand, there are approaches which transform the inertial information into an image-based representation to fed a CNN, taking advantage of its capabilities to work with images. Thus, Zhong and Deng [9] and Zhao and Zhou [11] transform the inertial signals in spaced time series, called Gait Dynamics Image (GDI), which are used as CNN input samples.

Traditionally, the cycle-stationary character of the march has been used to split the data into small subsequences. This helps to perform a faster and less expensive processing as the amount of input data is smaller than when a full sequence is employed. Thus, the full gait sequence is further subdivided using a cycle-based segmentation [21], [47]–[49] or a window-based segmentation [45], [50]. The former explicitly studies this previously mentioned cyclic character and creates a precise but complex segmentation. The latter, which is the simplest option, obtains the resulting segmentation following the assumption that one window should contain, at least, one complete gait cycle. Normally, the length of this window is between 1.4 [51] and 10 seconds [45].

In addition to using gait as an approach to *identify* people, it can be also applied to other tasks such as *gender* recognition or *age* estimation. Usually these tasks are independent of the main task [8], [52], but it has been demonstrated, like in applications for face recognition, that they can help to improve the results of the main task [53].

When different types of input data are available, fusion techniques can be used to improve the performance of the processing applied to those data. Two main methods can be employed for data fusion: early fusion and late fusion. On the one hand, early fusion methods, also known as feature fusion methods [54], [55], take data from multiple sensors and produce different features, which are merged at some stage of the pipeline to build a combined descriptor. On the other hand, late fusion methods, also known as decision fusion methods [56], fuse the output of independent classifiers applying some kind of arithmetic operations. Another option is explored in [34] and [57] where early and late fusion are applied together.

In this work, we develop an end-to-end approach based on a CNN which uses raw inertial data as input. We also propose the extension of our model with a fusion scheme that assumes the availability of multiple sensors generating different types of input data. In addition, we present a multi-task approach that, using a single model, improves the learning process by dealing with several tasks at the same time. Finally, we combine the fusion and the multi-task approach in a common model which uses multiple sensors to produce multiple outputs. To the best of our knowledge this is the first work that, employing inertial data, combines multiple inputs and multiple outputs in a single CNN to cope with the gait recognition problem.

III. PROPOSED APPROACH

A. PROBLEM DEFINITION

We propose an end-to-end approach based on Convolutional Neural Networks (CNNs) which automatically extracts discriminant features from a gait sequence. This kind of networks are suitable for this problem since they are based on convolutions which is a generic operation that can be used for any kind of signal. The proposed CNN uses the raw data acquired directly from the inertial sensors without any pre-processing step. The use of pre-processed features implies that a human has designed a set of operations to compute features according to his/her intuition about what is better to represent the input information. However, in our opinion, these pre-procesed features may not be the best possible ones since the human may not know all possible representations. On the other hand, using raw data as input to a CNN allows the model to learn its own features through backpropagation, automatically discovering the best features that maximize the accuracy on the target task.

Since we are going to use OU-ISIR Dataset [21] in our experiments, we propose a set of improvements for our model to take advantage of the information included in the dataset. This dataset includes information from multiple inertial sensors (*i.e.* accelerometer and gyroscope), thus, we plan to combine the information from all of them so that the model can build better features and, consequently, improve the global accuracy. In addition, as it also includes three labels per sample with information about *identity*, *age* and *gender* of a subject, our model processes the gait information in a multi-task setup to jointly recognize all of them. For more details about the dataset, the reader is referred to Sec. IV-A. Note that, although we focus on that dataset, our approach is applicable to any dataset that contains one or more sensors and labels for one or more characteristics of the subjects (*i.e.* id, gender, age, etc.). Finally, an identity verification (or *authentication*) system has also been implemented to decide if two different samples belong to the same subject.

We will use the following nomenclature throughout this paper, where vectors and matrices are marked in bold:

- $\mathbf{S}$: input sequence. It is composed by a temporal sequence of D channel measurements taken by a sensor.
- $\mathbf{s_i}$: i-th input sample consisting of a sub-sequence of $\mathbf{S}$ with a specific length, L. These fixed sized subsequences constitute the CNN inputs.
- y_i^t : label for sample $\mathbf{s_i}$ and task t.
- $g(\mathbf{s_i}, \theta)$: non-linear function applied to $\mathbf{s_i}$ with a set of parameters θ.
- $\widehat{\mathbf{y}}_\mathbf{i}$: output of the network for input $\mathbf{s_i}$.

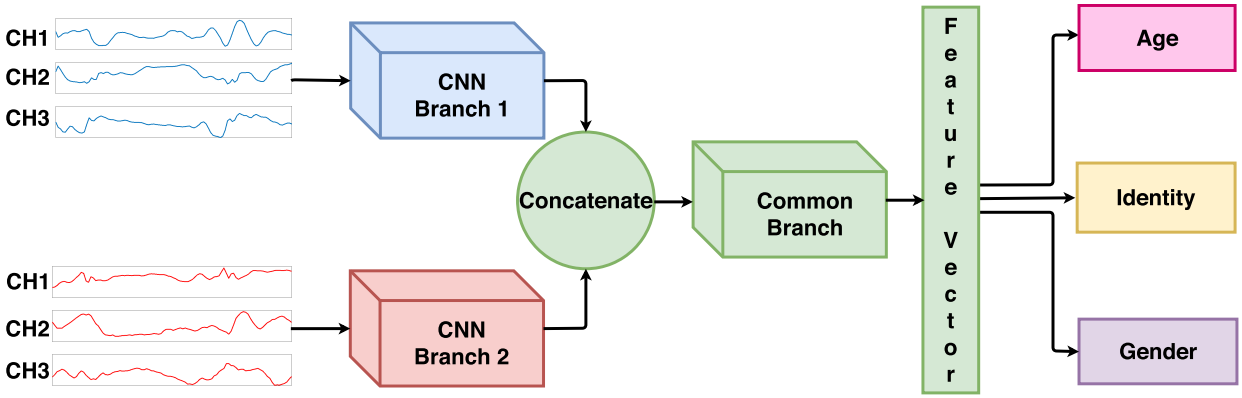

FIGURE 2. **Pipeline for multi-task and multi-sensor learning in a gait-based system. Each sensor has its own input branch to learn specific filters. At some point of the architecture, all branches are concatenated in a single Common Branch that produces a feature vector containing information from the multiple sensors. Finally, the prediction of the different tasks is carried out using the combined information.**

TABLE 1. **CNN architectures. Acronyms: 'P' = pooling size; 'Dr' = dropout; 'C' = number of classes for the used class; 'C_{id}' = number of classes for the *identification* task; 'C_{age}' = number of classes for the *age* task; 'C_{gender}' = number of classes for the *gender* task.**

CNN	*Conv01*	*Conv02*	*Conv03*	*Conv04*	*AvgPool*	*FC*
SingleTask SingleSensor	1 x 10 x 240 P: 1 x 2	1 x 7 x 300 P: 1 x 2	1 x 5 x 360 P: 1 x 2	1 x 3 x 420 P: 1 x 2	1 x 5 Dr: 0.5	C
MultiTask SingleSensor	1 x 10 x 240 P: 1 x 2	1 x 7 x 300 P: 1 x 2	1 x 5 x 360 P: 1 x 2	1 x 3 x 420 P: 1 x 2	1 x 5 Dr: 0.5	$C_{id} + C_{age} + C_{gender}$
SingleTask MultiSensor	Acc: 1 x 10 x 240 P: 1 x 2 Gyr: 1 x 10 x 240 P: 1 x 2	1 x 7 x 300 P: 1 x 2	1 x 5 x 360 P: 1 x 2	1 x 3 x 420 P: 1 x 2	1 x 5 Dr: 0.5	C
MultiTask MultiSensor	Acc: 1 x 10 x 240 P: 1 x 2 Gyr: 1 x 10 x 240 P: 1 x 2	1 x 7 x 300 P: 1 x 2	1 x 5 x 360 P: 1 x 2	1 x 3 x 420 P: 1 x 2	1 x 5 Dr: 0.5	$C_{id} + C_{age} + C_{gender}$

B. INITIAL CNN ARCHITECTURE

Since the CNN model requires a fixed input size and the length of the gait sequences is not equal for all of the subjects, we must normalize their length before processing them through the CNN. To take advantage of the characteristic stationary cycle of the human gait, we divide each sequence $\mathbf{S}$ into U subsequences $\mathbf{s_i}$, where $1 \leq i \leq U$, with a fixed and sufficient length to collect an entire gait cycle. Note that gait sequences are not divided into gait cycles, we just take subsequences with a fixed length from a sequence. Therefore, the sequence is divided into windows of length L where each dimension of the signal defines a specific input channel (leftmost part of Fig. 2 shows the input data considering $D = 3$ channels, which is also the number of physical dimensions). Thus, the convolutions of the first layer have a filter size of $1 \times N \times D$ to take advantage of the temporal information given by L (length of the sample) and the D dimensions (also called axes) supplied by sensor captures. Note that, in this case, N is the size of the convolution. To increment the number of subsequences available for training, windows are taken with an overlap of $O\%$.

We propose a CNN with four convolutional layers, whose number of filters gradually increases. ReLU, batch normalization and max pooling operations are added after each convolution. After the last convolutional layer, we change the max pooling operator by an average pooling one. Then, a dropout and a fully-connected (FC) layer, with as many outputs as available classes in the dataset, are appended. The final layer (*i.e.* output) is a Softmax one that obtains the probability distribution associated to the input sample.

The first row of Tab. 1, corresponding to the Single Task/Single Sensor case, describes the baseline architecture. Specifically, this table shows all convolutional and fully-connected layers together with the auxiliary layers attached to them (*i.e.* pooling layers size and dropout percentage). The line above each cell indicates the filter size and the number of filters. Thus, the first two numbers define the filter dimension, and the third one indicates the number of filters that comprises the layer. For example, $1 \times 10 \times 240$ represents a convolution with 240 filters of dimension 1×10. In the case of the pooling layers, shown in the bottom line of each table cell, the information indicates the window size that is applied by the pooling operators to the input data.

Note that the decision of using four layers of convolutions was taken after carrying out a battery of experiments with different number of layers and filters.

To train our model for the identification task (the multi-task loss function is explained in Sec. III-C), we use a cross-entropy loss which is commonly proposed to quantify the closeness between two probability distributions. It is defined by the following equation:

$$\mathcal{L}_m(\widehat{\mathbf{y}}, c) = -\widehat{y_c} + \log \sum_{k=1}^{K} e^{\widehat{y_k}}, \tag{1}$$

where $\widehat{y}$ is the output vector of the network, $\widehat{y_c}$ is the output for the target class, $\widehat{y_k}$ is the k-th component of the output vector, c is the ground-truth class and K is the total number of classes.

C. MULTI-TASK APPROACH

Since the dataset employed in this paper provides multiple type of labels, we also explore the application of deep multi-task models (DMT) to our problem. Training a deep multi-task model (DMT) with $T + 1$ tasks requires the use of a set of tuples $I = (\mathbf{s_i}, y_i^m, y_i^1, y_i^2, \ldots, y_i^T)$, where y_i^m is the label corresponding to the main task, and y_i^t, with $t \in [1, T]$, represents the label for each auxiliary task [7].

Taking into account the available labels in the chosen dataset, which include *identity*, *age* and *gender*, we define the following multi-task loss function $\mathcal{L}_{DMT}$ for a given sample $\mathbf{s}$:

$$\begin{aligned}\mathcal{L}_{DMT}(g(\mathbf{s}, \theta), \mathbf{Y}) = {} & \lambda_{\text{id}}\mathcal{L}_{\text{id}}(\widehat{\mathbf{y}}^{\text{id}}, y^{\text{id}}) \\ & + \lambda_{\text{age}}\mathcal{L}_{\text{age}}(\widehat{\mathbf{y}}^{\text{age}}, y^{\text{age}}) \\ & + \lambda_{\text{gender}}\mathcal{L}_{\text{gender}}(\widehat{\mathbf{y}}^{\text{gender}}, y^{\text{gender}}),\end{aligned} \tag{2}$$

where $Y = (y_i^{\text{id}}, y_i^{\text{age}}, y_i^{\text{gender}})$ and $\mathcal{L}_{\text{id}}$, $\mathcal{L}_{\text{age}}$, $\mathcal{L}_{\text{gender}}$ are the loss functions for *id*, *age* and *gender* tasks, respectively. Similarly, λ_{id}, λ_{age} and λ_{gender} are the weights given to the tasks, being λ_{id} equals to 1 (main task). For the other lambda values associated to the auxiliary tasks, we are going to use values between 0 and 1. In Section IV-D.3 we conduct some experiments to establish the most suitable values for each subtask.

Regarding the loss function used for each task, *identity* and *gender* employ the well-known expression of cross-entropy loss indicated in equation 1. In the case of the *age* task, the dataset labelling only considers a set of ranges. Since this task can be formulated as a classification problem, we employ the same loss function than the previous tasks. Fig. 2 shows a sketch of our CNN. It can be observed that a common feature vector obtained from the Common Branch computes several outputs at the same time, *i.e. age*, *identity* and *gender* (rightmost part of the figure).

D. MODALITY FUSION

Due to the existence of different kinds of input data coming from different sensors, we have designed a network which combines these inputs to benefit the classification accuracy by learning new relationships between different kinds of input data. To allow the model learning combined features automatically, the information coming from each sensor is fed into an individual branch composed of a specific number of convolutional layers that will compute specific predictors for each sensor. In Fig. 2 we can see an example of a CNN with two branches (named CNN Branch 1 and CNN Branch 2), where each branch receives information from a different sensor. Finally, the descriptors resulting from both branches are concatenated to produce a joint feature vector that is passed to a common branch (*i.e.* Common Branch in Fig. 2) that extracts combined features for all sensors.

Since there are many possible layers in the architecture where the information of the input branches can be combined, we have selected the best layer by cross-validating the different setups and selecting the best one according to the accuracy metric. To limit the number of possible tests, we have restricted the experiments to convolutional or average pooling layers. The details about this experiment can be found in Sec. IV-D.4.

E. IDENTITY AUTHENTICATION

In this problem, an input test sample is compared against samples of the training set. The answer of the system should be a positive output for samples belonging to the same subject and a negative output for samples belonging to different subjects.

The authentication procedure starts by feeding the input sample to the CNN so that its feature vector is extracted. Similarly, the feature vectors of the training samples are also calculated in the same way. Since the feature vectors can be obtained from any layer of the model, we perform a cross-validation process, which is explained in Sec. IV-E, to find out the layer that produces the best possible features. Once the features are extracted from the network, they are normalized with a L2-norm:

$$\mathbf{f_{norm}} = \frac{\mathbf{f}}{\sqrt{\sum_{i=1}^{n} |f_i|^2}}, \tag{3}$$

where $\mathbf{f_{norm}}$ is the normalized feature vector, $\mathbf{f}$ is the feature vector extracted from a specific layer, f_i is the i-th feature of $\mathbf{f}$ and n is the dimensionality of the feature vector.

Then, a distance vector $\mathbf{d}$ with as many components as samples we have in the training set is computed. The value of the i-th component of this vector is calculated by applying the Euclidean distance between the input sample and the j-th training sample.

Finally, in order to compute the Area Under Curve (AUC) or the Equal-Error-Rate (EER), we transform these distances into probabilities. With this aim, we propose an expression that normalizes the distance vector as follows:

$$\mathbf{probs} = 1 - \frac{\mathbf{d}}{\max(\mathbf{d})}, \tag{4}$$

where $\mathbf{probs}$ is the resulting probability vector for the input sample, $\mathbf{d}$ defines the distance vector from each test sample to each training sample and max is a function that extracts the maximum value of vector $\mathbf{d}$.

IV. EXPERIMENTS AND RESULTS

A. DATASET

The OU-ISIR dataset [21] is regarded as the largest gait dataset based on inertial sensors. The information is collected using one smartphone and three IMU sensors located in the waist of the subjects. An IMU sensor is an electronic device that measures the speed, orientation and gravitational forces of a person, animal of object that is moving. To obtain these measurements, the IMU sensor uses a combination of accelerometers and gyroscopes. The central IMU and the smartphone are located in the center back waist and the other two IMU sensors are placed on the left and right waist of the person, respectively.

The dataset is split into two subsets, the first one (part A), which is composed of 744 subjects (389 males and 355 females) with ages between 2 and 78 years, is recorded only with the central IMU sensor. In this subset, two sequences of data per person have been recorded at a rate of 100Hz. Following the methodology used in similar works ([9], [11], [12], [21], [39], [40], [42], [58]), the first sequence is used for training and the second one for testing. Three labels, namely *identity*, *gender* and *age*, are provided for each subject. Notice that *age* identification problem has been tackled using a classification approach, since the labels provided by the dataset only identify ranges of ages: 'Under10', 'Group10-19', 'Group20-29', 'Group30-39', 'Group40-49' and 'Over50'. The second subset (part B) is composed of 495 subjects recorded with the three IMU sensors. For each subject and each sensor, there are two sequences for *level* walk, a sequence for *up-slope* walk, and a sequence for *down-slope* walk.

Fig. 3 shows an example of the data acquisition process. The inertial data collected by the sensors is shown on the right part of the image. Top plot shows the temporal measurements for the accelerometer and bottom plot contains the measurements for the gyroscope. The image of the subject and the images of the signals have been obtained from Ngo *et al.* [21]. In our experiments we use the information from the accelerometer and gyroscope as independent signals that could be fused. Each sensor is composed of three signals, or axes, that represent a measurement in a three-dimensional space (*i.e.* X, Y and Z). Both sensors are oriented in the same direction, so their axes are equal.

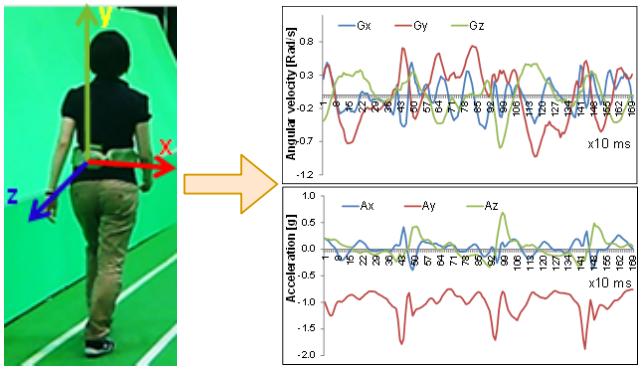

FIGURE 3. Data acquisition process. The left part of the image shows a subject walking through a circuit with the sensor system in the waist. The right part of the image depicts the inertial information recorded during the walk. Top plot shows the measurements for the accelerometer and bottom plot contains the measurements for the gyroscope. Images obtained from [21].

B. INPUT DATA

Due to the reduced amount of data available for training, we perform a data augmentation process before the training step. This way, from every original sequence, we obtain three new sequences applying data augmentation techniques. Concretely, we apply the following operations:

- Adding gaussian noise with $\sigma = 0.01$ to the input signal.
- Scaling the original sequence by a random value in the range 0.7 and 1.1 following Eq. 5.

$$\mathcal{L}_r(\mathbf{S}) = \mathbf{S} \cdot ((1.1 - 0.7) * rand() + 0.7), \qquad (5)$$

where $\mathbf{S}$ is the original signal and $rand()$ is a function that returns a random number extracted from an uniform distribution in the range [0, 1].

- Interpolating the original signal by inserting 10 new values between each pair of original values, and then randomly sampling this new signal, so that for each set of 10 values a single value is taken randomly. This way, sampling imprecision resulting from delays in the physical sensor or in its firmware is simulated.

With these techniques, the training process is improved and the network will be able to deal with different types of noise. Since the input sequences have different lengths, we split sequences into fixed-length subsequences of $L = 100$ measures (*i.e.* one second), what is enough to contain a complete gait cycle. In addition, to increase the number of samples even more, subsequences are extracted with an overlap of $O = 75\%$.

C. IMPLEMENTATION DETAILS

We ran our experiments on a computer with two Xeon E5-2698 processors, with 16 cores each one, running at 2.3GHz, 256 GB of RAM and a GPU NVidia Titan X Pascal, with MatConvNet library [59] running on Matlab 2016a for Ubuntu 16.04.

During training, we perform 100 epochs with a training set split into training + validation. Training process is finished with a fine-tuning step that runs 50 additional epochs with the whole training set, without validation. The learning rate starts at 0.01, and is divided by 10 every 50 epochs. The same policy is used in the case of fine-tuning, except that the starting rate is 0.001. We train the networks using standard stochastic gradient descent (SGD) with mini-batches of 128 samples, weight decay of 0.0005 and momentum of 0.9. Filters are initialized with random values from a normal distribution with zero-mean and standard derivation of 0.01. Bias are initialized to zero. Note that these hyper-parameters have been

TABLE 2. Gait recognition accuracy and F1-score of three different sensor positions. The best results are marked in bold.

IMU Position	Acc	F1-score
center	92.3	90.8
left	**95.2**	**94.0**
right	91.5	90.0

cross-validated before running the experiments presented in the following sections.

Since the input sequences are split into subsequences, at test time we have to combine the outputs of each subsequence to obtain a global accuracy for the whole sequence. With this aim, we combine the probabilities of all subsequences by multiplying them to obtain a final probability distribution using the following equation:

$$P(S = c) = \prod_{i=1}^{U} P_i(s_i = c), \tag{6}$$

where U is the number of subsequences extracted from sequence S, $P(S = c)$ is the probability of assigning the identity c to the person in sequence S and $P_i(s_i = c)$ is the probability of assigning the identity c to the person in subsequence s_i. Note that we performed a set of experiments with different strategies to combine the probabilities and we selected the best one (*i.e.* product of the probabilities).

D. GAIT RECOGNITION EXPERIMENTS

1) SENSOR POSITION

In this experiment we evaluate the performance of our baseline network (first row in Tab. 1) for three different positions of the accelerometer included in OU-ISIR part B (*i.e.* left, center and right waist).

Tab. 2 summarizes the accuracy and F1-score results for the identification problem using our baseline network with three different sensor positions. According to the results, the most discriminant position is the left waist while the other two positions obtain similar results. These results correlate with those obtained in [21].

Note that although the best accuracy has been obtained when the sensor is located at the left waist, the experiments conducted in the following sections will use the sensor at the center back waist as the most populated dataset (OU-ISIR part A) only provides information for this position. In addition, the state-of-the-art approaches, which we want to compare with, only use this subset.

2) SINGLE TASK WITH INDIVIDUAL SENSORS

In this experiment we compare the accuracy of each individual sensor to obtain the baseline results that will be used to evaluate the impact of the fusion and multi-task techniques.

Since there are three different labels or tasks per subject (*i.e. identity*, *age* and *gender*) and two sensors (*i.e.* accelerometer and gyroscope), we train as many networks as combinations are available. Thus, we train six CNNs following the specifications commented in Sec. III and the architecture defined in the first row of Tab. 1. Note that for all these networks, our input data shape is $1 \times 100 \times 3$.

Following the experimental setup commented in Sec. IV-A, the first sequence of each subject is used for training and the second one for testing. In addition, we perform the data augmentation and operations commented in Sec. IV-B.

The first two rows in Tab. 3 summarizes the accuracy and the F1-score results for this experiment. We observe that the accelerometer obtains the best results for all tasks although the difference with the gyroscope is always lower than 1%. Thus, this means that both sensors can be used to recognize people using their way of walking. As the results obtained with both sensors are high, we expect the fusion of both input data can boost the results.

3) MULTI-TASK WITH INDIVIDUAL SENSORS

The objective of this experiment is to validate if a multi-task training process could improve the baseline results. Thus, we train one network per sensor with a multi-task loss as explained in Sec. III-C. The architecture used for these networks is defined in the second row of Tab. 1. According to Eq. 2, the value taken by the lambda parameter λ_{id}, belonging to the main task is 1.0, while the corresponding ones for auxiliary tasks are problem-dependant. Thus, we have selected the best values for these parameters through cross-validation running a set of experiments with values ranging from 0.1 to 1.0. After this cross-validation process, the best values obtained for each lambda are 0.6 and 0.7 for *age* and *gender* respectively. Note that for all these networks, our input data shape is $1 \times 100 \times 3$ and three labels are employed, one per task.

Third and fourth rows of Tab. 3 show the results for this experiment. Again, the accelerometer achieves the best results for all tasks compared to gyroscope. Comparing the multi-task results for each sensor with the baseline results, we can observe that the multi-task loss improves the accuracy and F1-score for all tasks and sensors. On average ('Avg' columns in Tab. 3), the improvement is 1.6% for the accelerometer and 0.7% for the gyroscope. Therefore, the multi-task loss helps the optimization process to find better descriptors and, consequently, improves the global performance of the model.

Apart from the improvement in the results, the multi-task model also has an important impact in computing performance. Thus, it is able to produce the output for all tasks at the same time using the same parameters. This implies a saving of time in both training and testing as only one model must be computed instead of three models if a single task setup had been implemented.

4) SELECTION OF THE FUSION POSITION

To deal with the information from both sensors, we use a CNN with two branches, one per sensor, that will be fused at some point of the architecture. Note that, as explained above

TABLE 3. Gait recognition accuracy and F1-score. 'Avg' corresponds to the average accuracy or F1-score of the three tasks. The best results are marked in bold.

	Acc				F1-score			
Architecture	*Id*	*Age*	*Gender*	*Avg*	*Id*	*Age*	*Gender*	*Avg*
SingleTask Accelerometer	89.7	91.0	94.8	91.8	87.6	91.3	94.5	91.2
SingleTask Gyroscope	89.1	89.1	94.4	90.9	87.5	89.7	94.4	90.5
MultiTask Accelerometer	90.9	93.3	95.9	93.4	89.1	93.3	95.9	92.8
MultiTask Gyroscope	90.1	90.1	94.8	91.7	88.3	90.5	94.9	91.2
SingleTask Fusion	94.2	95.0	95.6	94.9	93.5	95.0	95.6	94.7
MultiTask Fusion	**94.8**	**96.1**	**97.7**	**96.2**	**93.8**	**96.3**	**97.7**	**95.9**

TABLE 4. CNN architectures for fusion experiments. Acronyms: 'P' = pooling size; 'Dr' = dropout; 'C_{id}' = number of outputs in identification task (744 outputs). Each row represents an architecture that performs the early fusion in a different layer and each column represents the different layers that make up these neural networks.

Fusion Position	*Conv01*	*Conv02*	*Conv03*	*Conv04*	*AvgPool*	*FC*
Conv01 CNN	Acc: 1 x 10 x 240 P: 1 x 2 Gyr: 1 x 10 x 240 P: 1 x 2	1 x 7 x 300 P: 1 x 2	1 x 5 x 360 P: 1 x 2	1 x 3 x 420 P: 1 x 2	1 x 5 Dr: 0.5	C_{id}
Conv02 CNN	Acc: 1 x 10 x 240 P: 1 x 2 Gyr: 1 x 10 x 240 P: 1 x 2	Acc: 1 x 7 x 300 P: 1 x 2 Gyr: 1 x 7 x 300 P: 1 x 2	1 x 5 x 360 P: 1 x 2	1 x 3 x 420 P: 1 x 2	1 x 5 Dr: 0.5	C_{id}
Conv03 CNN	Acc: 1 x 10 x 240 P: 1 x 2 Gyr: 1 x 10 x 240 P: 1 x 2	Acc: 1 x 7 x 300 P: 1 x 2 Gyr: 1 x 7 x 300 P: 1 x 2	Acc: 1 x 5 x 360 P: 1 x 2 Gyr: 1 x 5 x 360 P: 1 x 2	1 x 3 x 420 P: 1 x 2	1 x 5 Dr: 0.5	C_{id}
Conv04 CNN	Acc: 1 x 10 x 240 P: 1 x 2 Gyr: 1 x 10 x 240 P: 1 x 2	Acc: 1 x 7 x 300 P: 1 x 2 Gyr: 1 x 7 x 300 P: 1 x 2	Acc: 1 x 5 x 360 P: 1 x 2 Gyr: 1 x 5 x 360 P: 1 x 2	Acc: 1 x 3 x 420 P: 1 x 2 Gyr: 1 x 3 x 420 P: 1 x 2	1 x 5 Dr: 0.5	C_{id}
AvgPool CNN	Acc: 1 x 10 x 240 P: 1 x 2 Gyr: 1 x 10 x 240 P: 1 x 2	Acc: 1 x 7 x 300 P: 1 x 2 Gyr: 1 x 7 x 300 P: 1 x 2	Acc: 1 x 5 x 360 P: 1 x 2 Gyr: 1 x 5 x 360 P: 1 x 2	Acc: 1 x 3 x 420 P: 1 x 2 Gyr: 1 x 3 x 420 P: 1 x 2	Acc: 1 x 5 Dr: 0.5 Gyr: 1 x 5 Dr: 0.5	C_{id}

in Sec. III-D, our fusion is performed as the concatenation of the embedding of the different inputs.

Since the proposed architecture is composed of multiple layers, data fusion can be performed at each one of them. In order to find out the best layer to fuse, we perform a set of experiments where branches are fused after each convolutional layer of the network and after the average pooling layer. As we are dealing with three tasks, two different networks and five fusion positions, the number of experiments to be carried out is be very large. Thus, to reduce the experiments, we restrict the study to only the identification task. Then, the results will be extrapolated to the rest of the tasks and to the multi-task approach. Tab. 4 shows the architectures used during this experiment. Each row represents a different architecture and each column indicates the configuration of each layer of the architecture. When a filter is preceded by *Acc* or *Gyr* it means that this filter is applied to data from the accelerometer or the gyroscope, respectively. Otherwise, it is applied to the fused data.

Tab. 5 contains the comparative results for the different fusion positions. As we can see, both the accuracy and F1-score drop as fusion is deeper applied in the architecture.

TABLE 5. Fusion level experiment. Each row represents a different layer where the fusion is applied. 'Acc' column represents the accuracy and 'F1-score' column represents the F1-score for the identification problem, respectively.

Position	*Acc*	F1-score
Conv01	**94.2**	**92.9**
Conv02	94.0	92.7
Conv03	93.4	92.0
Conv04	89.0	86.7
AvgPool	89.0	86.7

Thus, the best result is obtained when fusing just after the first convolution while the worst one is attained after the last convolution and after the average pooling. We think this behaviour is due to the fact that the training dataset is relatively small and, when the fusion is performed in later layers, there are more trainable parameters and the models overfit. Probably, with a bigger dataset, the best results would be obtained in later layers where the stored knowledge is more discriminant. Unfortunately, currently there are no larger datasets. Consequently, fusion after the first convolutional layer will be employed in the following experiments.

5) SINGLE TASK WITH FUSION

In this experiment we focus on the fusion approach to check its impact on the accuracy compared to the baseline. Taking advantage of the findings obtained in the previous experiment, we design a network with two branches that will be combined after the first convolutional layer.

Since there are three different labels or tasks per subject (*i.e. identity*, *age* and *gender*), we train three CNNs, one per task, following the specifications commented in Sec. III and the architecture defined in the third row of Tab. 1. Note that for all these networks, our input data shape is $1 \times 100 \times 3$ for each branch.

The fifth row of Tab. 3 contains the results for this experiment. As we can observe, the fusion improves the baseline results for all tasks and, on average, the improvement is around a 3% compared to the accelerometer, which, attending to accuracy and F1-score results, is the best sensor. If we compare the fusion approach with the best multi-task sensor, we can see that the results are better for *id* and *age* but a bit worse for *gender*. However, on average, fusion improves in a 1.3% the best results of multi-task in terms of accuracy and in a 1.9% for F1-score.

6) MULTI-TASK WITH FUSION

Finally, we evaluate the effect of applying fusion and multi-task in the same network. Therefore, a network with two branches is designed, which are combined after the first convolutional layer. Also, a multi-task loss for training the model is included.

In this case, we only have to train one CNN as all tasks are used at the same time. We follow the specifications commented in Sec. III and the architecture defined in the last row of Tab. 1. Note that our input data shape is $1 \times 100 \times 3$ for each branch. An sketch of this model can be seen in Fig. 2. Note that in this experiment, our lambda values are 0.9 and 0.8 for *age* and *gender*, respectively.

The last row in Tab. 3 contains the results for this experiment. According to the results, this approach is the best one compared with the other architectures of the previous experiments. These results validate our hypothesis claiming that the use of more modalities and labels boosts the results as the model has more information to describe the subjects. On average, the model with multi-task and fusion achieves an improvement of 4.4% for accuracy and 4.7% for F1-score with respect to the accelerometer without multi-task.

Moreover, this model is able to produce outputs for all tasks at the same time using the same parameters. This means a saving of time both in training and in test as we have to deal with only one model instead of three when a single task setup is used.

E. AUTHENTICATION EXPERIMENTS

In these experiments, the evaluation of our models is performed following the indications commented in Sec. III-E. Thus, we extract the activations of the average pooling layer for each sample to be tested and compute the euclidean distance between it and the training samples. Then, we evaluate the performance of our approach calculating the Area Under the Curve (AUC) and the Equal Error Rate (EER) metrics, which have been previously extracted from a ROC curve.

TABLE 6. Authentication accuracy. 'EER' is the Equal-Error-Rate (lower is better) and 'AUC' is the Area Under the Curve (higher is better). The best results are marked in bold.

Architecture	*EER*	*AUC*
SingleTask Accelerometer	1.47	99.91
SingleTask Gyroscope	2.50	99.80
MultiTask Accelerometer	1.61	99.90
MultiTask Gyroscope	2.85	99.72
SingleTask Fusion	**1.14**	**99.93**
MultiTask Fusion	1.34	99.92

Tab. 6 shows the AUC and EER for our different models defined in the recognition experiments. Thus, each row is a different model and each column is a different metric. Comparing the results appearing in the table for single sensors (first and second rows), the accelerometer obtains the best results for both metrics compared to the gyroscope. If we focus on the MultiTask experiments (third and fourth rows), we can see that the results are slightly worse than the obtained with the isolated sensors, specially for the EER metric. In our opinion, this worsening of the results is due to the multi-task learning setup as the model has to deal with different labels. Thus, the descriptors used for the authentication process contains information used for a different task and the authentication gets worse precision. Finally, if we focus on the fusion experiments (rows fifth and sixth), the results improve on the previous ones but, again, the multi-task learning obtains worse results than the single-task learning.

F. STATE OF THE ART COMPARISON

In this section, we compare our best approaches for recognition and authentication with the state-of-the-art. Note that the works included in the comparison are explained in Sec. II. Tab. 7 contains the results for the recognition problem and Tab. 8 contains the results for the authentication problem.

In both cases, our approach sets a new state-of-the-art by a wide margin. For the recognition problem, our MultiTask Fusion improves the previous approach by a 11% for the identification task. Note that the other approaches do not use other tasks ('-' values in the table) so the comparison for those cases cannot be performed. For the authentication problem, our SingleTask Fusion obtains the best results and improves by 4.5% the previous best EER result. Therefore, our end-to-end method is able to beat all the previous approaches, establishing a new state-of-the-art.

G. EXECUTION TIME DURING TEST

As stated by the results reported in the previous sections, the use of a multi-task learning combined with the fusion of information from multiple sensors is able to improve the baseline model and previous works in terms of accuracy.

TABLE 7. Gait recognition: state-of-the-art. Each row represents a different approach. *'Avg'* is the average accuracy of the three tasks. The best results are marked in bold.

CNN	*Id*	*Age*	*Gender*	*Avg*
AE-GDI-CNN [11]	61.0	-	-	-
Muaaz *et al.* [58]	63.5	-	-	-
Ngo *et al.* [21]	70.2	-	-	-
Wei *et al.* [12]	83.8	-	-	-
MultiTask Fusion (Ours)	**94.8**	**96.1**	**97.7**	**96.2**

TABLE 8. Authentication: state-of-the-art. Each row represents a different approach. 'EER' is the Equal-Error-Rate. The best results are marked in bold.

Approach	*EER*
Gafurov *et al.* [39]	15.8
Derawi *et al.* [42]	14.3
Rong *et al.* [40]	14.3
Ngo *et al.* [21]	13.5
Inp GDI + i-vector [9]	7.1
NC GDI + i-vector [9]	5.6
SingleTask Fusion (Ours)	**1.1**

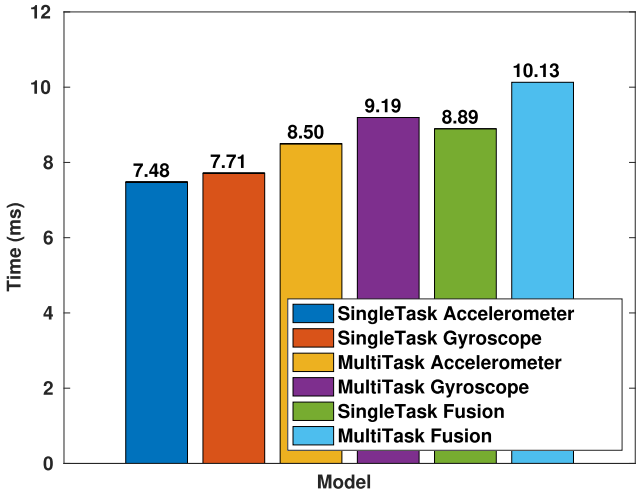

FIGURE 4. Inference time comparison. Execution time (ms) of the proposed models for inference.

In this section, we evaluate the execution time of our models during the test process (inference). Experiments have been conducted on a heterogeneous architectures consisting of a host with two Intel Xeon E5-2698 CPUs and a NVIDIA GeForce GTX 980 device connected to the host through a PCIe 3.0 bus.

In our experiments, the transfer time of an input sample from CPU main memory to GPU global memory through the PCIe bus is negligible (around 1μs). Then, the sample is processed on the GPU and the execution time for the inference is calculated. The experiment for each CNN model is run ten times and the average time is calculated. Obtained results are shown in Fig. 4. According to the results, the baseline networks (*i.e.* SingleTask models) are the fastest ones. When fusion or multi-task are included, the computation increases and therefore the execution time increases too. For SingleTask and MultiTask models, the execution time increases approximately in 1 ms but the accuracy of the model increases in more than 3%. Regarding the MultiTask models, it is important to point out that despite they have a higher execution time than SingleTask models, they produce three simultaneous outputs (id, age and gender). However, SingleTask models should be execute three times to obtain the same outputs. Consequently, the MultiTask models are the fastest ones when multiple outputs are necessary and, in addition, they produce the most accurate results.

V. CONCLUSIONS

We have presented a new end-to-end approach based on CNN architectures for the gait-based recognition and authentication problems that uses raw inertial data as input. A fusion scheme has also been proposed which takes advantage of data obtained from several inertial sensors. In addition, we have developed a multi-task learning model that works with the multiple labels of the dataset. Extensive cross-validation has been employed to establish the best hyper-parameter values of the models, such as the layer to fuse, λ values, etc.

As a result, the proposed architectures are able to extract automatically gait features from sequences of inertial information recorded by accelerometers or gyroscopes. Those signatures have been used in four different tasks: people *identification*, *gender* recognition, *age* recognition and people *authentication*. In all cases, our approach sets a new state-of-the-art compared to previous approaches.

With regard to the input sensor, according to our results, the accelerometer obtains the best results in all setups. However, the gyroscope achieves results which are close to the ones obtained with the accelerometer. The fusion of both sensors improves the accuracy in all cases showing that the use of multiple inputs benefits the learning process. Regarding the multi-task problem, we have demonstrated that a single model can be trained in a multi-task setup to obtain the outputs of all tasks at the same time, instead of having one model per task. Moreover, the multi-task learning boosts the results for the tasks of people *identification*, *gender* recognition and *age* recognition. In the case of people *authentication*, the single task models obtain better results although by a small margin.

As a final recommendation and, according to the results obtained, the best option would be to use an end-to-end approach fusing the information from all sensors. Multi-task learning can improve the results for recognition problems but, for *authentication*, the performance is a bit lower compared to a single task approach.

As future work, we plan to study in detail the multi-task setup including *authentication* to improve its performance. We intuit that including a verification loss during training should be a good starting point to improve the current results. In addition, we plan to study how the gait is affected by illness or fatigue in terms of recognition accuracy.

The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

REFERENCES

[1] M. P. Murray, A. B. Drought, and R. C. Kory, "Walking patterns of normal men," *J. Bone Joint Surg.*, vol. 46, no. 2, pp. 335–360, 1964.

[2] M. J. O'Malley, M. F. Abel, D. L. Damiano, and C. L. Vaughan, "Fuzzy clustering of children with cerebral palsy based on temporal-distance gait parameters," *IEEE Trans. Rehabil. Eng.*, vol. 5, no. 4, pp. 300–309, Dec. 1997.

[3] R. de Melo Roiz *et al.*, "Gait analysis comparing Parkinson's disease with healthy elderly subjects," *Arquivos Neuro-Psiquiatria*, vol. 68, pp. 81–86, Feb. 2010.

[4] K. Jellinger, D. Armstrong, H. Y. Zoghbi, and A. K. Percy, "Neuropathology of rett syndrome," *Acta Neuropathol.*, vol. 76, no. 2, pp. 142–158, Mar. 1988.

[5] D. Gafurov, "A survey of biometric gait recognition: Approaches, security and challenges," in *Proc. Annu. Norwegian Comput. Sci. Conf.*, 2007, pp. 19–21.

[6] M. Bächlin, J. Schumm, D. Roggen, and G. Töster, "Quantifying gait similarity: User authentication and real-world challenge," in *Advances in Biometrics*, M. Tistarelli and M. S. Nixon, Eds. Berlin, Germany: Springer, 2009, pp. 1040–1049.

[7] M. J. Marín-Jiménez, F. M. Castro, N. Guil, F. de la Torre, and R. Medina-Carnicer, "Deep multi-task learning for gait-based biometrics," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 106–110.

[8] F. M. Castro, M. J. Marín-Jiménez, and N. Guil, "Multimodal features fusion for gait, gender and shoes recognition," *Mach. Vis. Appl.*, vol. 27, no. 8, pp. 1213–1228, Nov. 2016.

[9] Y. Zhong and Y. Deng, "Sensor orientation invariant mobile gait biometrics," in *Proc. IEEE Int. Joint Conf. Biometrics*, Sep./Oct. 2014, pp. 1–8.

[10] S. Sprager and M. B. Juric, "An efficient HOS-based gait authentication of accelerometer data," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 7, pp. 1486–1498, Jul. 2015.

[11] Y. Zhao and S. Zhou, "Wearable device-based gait recognition using angle embedded gait dynamic images and a convolutional neural network," in *Sensors*, vol. 17, no. 3, p. 478, 2017.

[12] Z. Wei, W. Qinghui, D. Muqing, and L. Yiqi, "A new inertial sensor-based gait recognition method via deterministic learning," in *Proc. 34th Chin. Control Conf. (CCC)*, Jul. 2015, pp. 3908–3913.

[13] M. Gadaleta and M. Rossi. (Oct. 2016). "IDNet: Smartphone-based gait recognition with convolutional neural networks." [Online]. Available: https://arxiv.org/abs/1606.03238

[14] J. Wang, M. She, S. Nahavandi, and A. Kouzani, "A review of vision-based gait recognition methods for human identification," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl.*, Dec. 2010, pp. 320–327.

[15] J. L. Moing and I. Stengel, "The smartphone as a gait recognition device impact of selected parameters on gait recognition," in *Proc. Int. Conf. Inf. Syst. Secur. Privacy (ICISSP)*, Feb. 2015, pp. 322–328.

[16] A. H. Johnston and G. M. Weiss, "Smartwatch-based biometric gait recognition," in *Proc. IEEE 7th Int. Conf. Biometrics Theory, Appl. Syst. (BTAS)*, Sep. 2015, pp. 1–6.

[17] J. Paefgen, F. Kehr, Y. Zhai, and F. Michahelles, "Driving behavior analysis with smartphones: Insights from a controlled field study," in *Proc. MUM*, 2012, Art. no. 36.

[18] N. Noury *et al.*, "Fall detection—Principles and methods," in *Proc. 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2007, pp. 1663–1666.

[19] Z. Li and G. Zhang, "A gait recognition system for rehabilitation based on wearable micro inertial measurement unit," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Dec. 2011, pp. 1678–1682.

[20] M. Derawi and P. Bours, "Gait and activity recognition using commercial phones," *Comput. Secur.*, vol. 39, pp. 137–144, Nov. 2013.

[21] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi, "The largest inertial sensor-based gait database and performance evaluation of gait-based personal authentication," *Pattern Recognit.*, vol. 47, no. 1, pp. 228–237, 2014.

[22] K. Van Laerhoven and O. Cakmakci, "What shall we teach our pants?" in *Proc. 4th IEEE Int. Symp. Wearable Comput. (ISWC)*, Oct. 2000, pp. 77–83.

[23] A. T. Özdemir and B. Barshan, "Detecting falls with wearable sensors using machine learning techniques," *Sensors*, vol. 14, no. 6, pp. 10691–10708, 2014.

[24] A. Samà *et al.*, "Dyskinesia and motor state detection in parkinson's disease patients with a single movement sensor," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Sep. 2012, pp. 1194–1197.

[25] S. Herrlich *et al.*, *Ambulatory Treatment and Telemonitoring of Patients With Parkinson's Disease*. Berlin, Germany: Springer, 2011, pp. 295–305.

[26] H.-C. Chang, Y.-L. Hsu, S.-C. Yang, J.-C. Lin, and Z.-H. Wu, "A wearable inertial measurement system with complementary filter for gait analysis of patients with stroke or Parkinson's disease," *IEEE Access*, vol. 4, pp. 8442–8453, 2016.

[27] M. Alotaibi and A. Mahmood, "Improved gait recognition based on specialized deep convolutional neural networks," in *Proc. IEEE Appl. Imag. Pattern Recognit. Workshop (AIPR)*, Oct. 2015, pp. 1–7.

[28] H. Su and F.-G. Huang, "Human gait recognition based on motion analysis," in *Proc. Int. Conf. Mach. Learn. Cybern.*, vol. 7, Aug. 2005, pp. 4464–4468.

[29] F. M. Castro, M. J. Marín-Jimènez, N. G. Mata, and R. Muñoz-Salinas, "Fisher motion descriptor for multiview gait recognition," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 31, no. 01, p. 1756002, 2017.

[30] M. J. Marín-Jiménez, F. M. Castro, Á. Carmona-Poyato, and N. Guil, "On how to improve tracklet-based gait recognition systems," *Pattern Recognit. Lett.*, vol. 68, pp. 103–110, Dec. 2015.

[31] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S.-M. Makela, and H. A. Ailisto, "Identifying users of portable devices from gait pattern with accelerometers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 2, Mar. 2005, pp. ii/973–ii/976.

[32] S. Šprager and D. Zazula, "A cumulant-based method for gait identification using accelerometer data with principal component analysis and support vector machine," *WSEAS Trans. Signal Process.*, vol. 5, no. 11, pp. 369–378, 2009.

[33] H. Sun and T. Yuao, "Curve aligning approach for gait authentication based on a wearable accelerometer," *Physiol. Meas.*, vol. 33, no. 6, pp. 1111–1120, 2012.

[34] O. Dehzangi, M. Taherisadr, and R. ChangalVala, "IMU-based gait recognition using convolutional neural networks and multi-sensor fusion," *Sensors*, vol. 17, no. 12, p. 2735, 2017.

[35] N. T. Trung, Y. Makihara, H. Nagahara, R. Sagawa, Y. Mukaigawa, and Y. Yagi, "Phase registration in a gallery improving gait authentication," in *Proc. Int. Joint Conf. Biometrics (IJCB)*, Oct. 2011, pp. 1–7.

[36] C. Shen, Y. Chen, and X. Guan, "Performance evaluation of implicit smartphones authentication via sensor-behavior analysis," *Inf. Sci.*, vols. 430–431, pp. 538–553, Mar. 2018.

[37] P. Connor and A. Ross, "Biometric recognition by gait: A survey of modalities and features," *Comput. Vis. Image Understand.*, vol. 167, pp. 1–27, Feb. 2018.

[38] N. V. Boulgouris, K. N. Plataniotis, and D. Hatzinakos, "Gait recognition using dynamic time warping," in *Proc. IEEE 6th Workshop Multimedia Signal Process.*, Sep./Oct. 2004, pp. 263–266.

[39] D. Gafurov, E. Snekkenes, and P. Bours, "Improved gait recognition performance using cycle matching," in *Proc. IEEE 24th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Apr. 2010, pp. 836–841.

[40] L. Rong, Z. Jianzhong, L. Ming, and H. Xiangfeng, "A wearable acceleration sensor system for gait recognition," in *Proc. 2nd IEEE Conf. Ind. Electron. Appl.*, May 2007, pp. 2654–2659.

[41] C. Nickel and C. Busch, "Classifying accelerometer data via hidden Markov models to authenticate people by the way they walk," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 28, no. 10, pp. 29–35, Oct. 2013.

[42] M. O. Derawi, P. Bours, and K. Holien, "Improved cycle detection for accelerometer based gait authentication," in *Proc. 6th Int. Conf. Intell. Inf. Hiding Multimedia Signal Process.*, Oct. 2010, pp. 312–317.

[43] Y. Watanabe, "Influence of holding smart phone for acceleration-based gait authentication," in *Proc. 5th Int. Conf. Emerg. Secur. Technol.*, Sep. 2014, pp. 30–33.

[44] S. Choi, I.-H. Youn, R. LeMay, S. Burns, and J.-H. Youn, "Biometric gait recognition based on wireless acceleration sensor using k-nearest neighbor classification," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2014, pp. 1091–1095.

[45] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Cell phone-based biometric identification," in *Proc. 4th IEEE Int. Conf. Biometrics, Theory, Appl. Syst. (BTAS)*, Sep. 2010, pp. 1–7.

[46] S. Khandelwal and N. Wickström, "Novel methodology for estimating initial contact events from accelerometers positioned at different body locations," *Gait Posture*, vol. 59, pp. 278–285, Jan. 2018.

[47] K. Sugandhi, F. F. Wahid, and G. Raju, "Detection of human gait cycle: An overlap based approach," in *Proc. Int. Conf. Infocom Technol. Unmanned Syst. (Trends Future Directions) (ICTUS)*, Dec. 2017, pp. 1–3.

[48] T. T. Ngo, Y. Makihara, H. Nagahara, Y. Mukaigawa, and Y. Yagi, "Similar gait action recognition using an inertial sensor," *Pattern Recognit.*, vol. 48, no. 4, pp. 1289–1301, 2015.

[49] P. Fernandez-Lopez, J. Sanchez-Casanova, P. Tirado-Martín, and J. Liu-Jimenez, "Optimizing resources on smartphone gait recognition," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2017, pp. 31–36.
[50] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes-Ortiz, *Human Activity Recognition on Smartphones Using a Multiclass Hardware-Friendly Support Vector Machine*. Berlin, Germany: Springer, 2012, pp. 216–223.
[51] T. Oberg, A. Karsznia, and K. Oberg, "Basic gait parameters: Reference data for normal subjects, 10–79 years of age," *J. Rehabil. Res. Develop.*, vol. 30, no. 2, pp. 210–223, 1993.
[52] M. Hofmann, J. Geiger, S. Bachmann, B. Schuller, and G. Rigoll, "The TUM gait from audio, image and depth (GAID) database: Multimodal recognition of subjects and traits," *J. Vis. Commun. Image Represent.*, vol. 25, no. 1, pp. 195–206, 2014.
[53] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. NIPS*, 2014, pp. 1988–1996.
[54] X. Wang, R. Bai, X. Cui, T. Wu, and Z. Qian, "Research on data fusion algorithm for attitude detection systems based on MEMS and magnetoresistive sensors," in *Proc. 9th Int. Conf. Adv. Infocomm Technol. (ICAIT)*, Nov. 2017, pp. 68–74.
[55] Q. Zou, L. Ni, Q. Wang, Q. Li, and S. Wang, "Robust gait recognition by integrating inertial and RGBD sensors," *IEEE Trans. Cybern.*, vol. 48, no. 4, pp. 1136–1150, Apr. 2018.
[56] L. Kaliciak, H. Myrhaug, A. Goker, and D. Song, "On the duality of specific early and late fusion strategies," in *Proc. 17th Int. Conf. Inf. Fusion (FUSION)*, Jul. 2014, pp. 1–8.
[57] Y. Chai, J. Ren, H. Zhao, Y. Li, J. Ren, and P. Murray, "Hierarchical and multi-featured fusion for effective gait recognition under variable scenarios," *Pattern Anal. Appl.*, vol. 19, no. 4, pp. 905–917, 2015.
[58] M. Muaaz and C. Nickel, "Influence of different walking speeds and surfaces on accelerometer-based biometric gait recognition," in *Proc. 35th Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2012, pp. 508–512.
[59] A. Vedaldi and K. Lenc, "MatConvNet: Convolutional neural networks for MATLAB," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 689–692.

RUBÉN DELGADO-ESCAÑO received the bachelor's degree in computer engineering from the University of Málaga, Málaga, Spain. His research interests include gait recognition, inertial information understanding, and machine learning.

FRANCISCO M. CASTRO received the bachelor's degree in computer science from the University of Córdoba, Córdoba, Spain. He is currently pursuing the Ph.D. degree with the University of Málaga. He was a Visiting Student with the THOTH Group, Inria, France. He is currently a Researcher with the University of Málaga. His research interests include human detection, gait recognition, human-centric video understanding, and machine learning.

JULIÁN RAMOS CÓZAR received the B.S. and Ph.D. degrees in telecommunication engineering from the University of Málaga, Málaga, Spain, in 1995 and 2002, respectively. He is currently an Associate Professor with the Department of Computer Architecture, University of Málaga. He has published more than 30 papers in international journals and conferences. His research interests include parallel computing, video and image processing, and machine learning.

MANUEL J. MARÍN-JIMÉNEZ received the B.Sc., M.Sc., and Ph.D. degrees from the University of Granada, Spain. He was a Visiting Student at the Computer Vision Center of Barcelona, Spain, Vislab-ISR/IST of Lisboa, Portugal, and the Visual Geometry Group of Oxford, U.K. He is currently an Associate Professor with the University of Córdoba, Spain. He has published more than 50 technical papers in journals and international conferences. His research interests include object detection, human-centric video understanding, and machine learning.

NICOLÁS GUIL received the B.S. degree in physics from the University of Sevilla, Spain, in 1986, and the Ph.D. degree in computer science from the University of Málaga, in 1995. He is currently a Full Professor with the Department of Computer Architecture, University of Málaga. He has published more than 80 papers in international journals and conferences. His research interests include the areas of parallel computing and video and image processing.