Journey into Gait Biometrics: Integrating Deep Learning for Enhanced Pattern Recognition

Anubha Parashar[a,*], Apoorva Parashar[b] and Imad Rida[e]

[a] School of Computing & Information Technology, Manipal University Jaipur, Jaipur, Rajasthan, India
[b] Consultant, Emerging technology, Mahindra Integrated Business Solutions, Mumbai, India,
[e] BMBI Laboratory, University of Technology of Compiègne, 60200, Compiègne, France

ARTICLE INFO

Keywords:
Gait Recognition
Biometrics
Deep Learning
Surveillance
Pattern Recognition

ABSTRACT

Exploring Gait Biometrics within the domain of deep learning offers a potent fusion that significantly enhances pattern recognition capabilities. Over the past decade, the evolution of deep learning (DL) pipelines has showcased their effectiveness in overcoming complex challenges within image and signal processing applications. Constructing these pipelines requires a deep understanding of the diverse intermediate layers and their implications. The iterative refinement process involves careful selection and rigorous performance validation of each configuration, demanding significant time and contemplation. Consequently, the task of selecting a robust DL pipeline that excels across various datasets remains challenging. The central objective of this review is to provide guidance to researchers, fostering a comprehensive grasp of distinct gait sensing technologies, while establishing a solid foundation in deep learning concepts. Although gait recognition is a relatively recent development and is yet to find widespread application in real-world scenarios, this article offers a thorough examination of gait biometrics tailored specifically for real-time surveillance applications. Delving into the complexities, it elucidates the crucial parameters governing deep learning pipelines and their nuanced selection to address specific challenges. Through an analysis of recent research articles on deep learning models and their performance across diverse datasets, the review outlines the merits and demerits of various approaches. The ultimate aim is to facilitate the development of an optimized pipeline that seamlessly integrates existing methodologies, enabling the attainment of swift yet precise results for a given problem.

1. Introduction

The rise in popularity of automated identification over the past few years has increased the focus of professionals working in computer vision and kinematics on gait recognition in real-time settings. This is because automated identification entries are becoming increasingly common. A person's gait, which can be recognized from a great distance and does not need cooperation from the target, is one of the most important biometric qualities humans possess. In vision-based techniques, it is likely that videos from low-resolution cameras will be carried out. Other biometric identification methods have absolutely no possibility of functioning correctly in such stated scenarios [76]. Thanks to this capability of gait, it is now possible to employ it in a real-time environment. Because gait patterns are challenging to replicate and substantially more difficult to hide than facial characteristics, they are regarded as more secure forms of biometric identification [77].

Deep learning has emerged as a promising method for recognizing humans through gait [80]. Starting gait recognition with deep learning might be difficult since researchers don't know which deep learning pipeline to use or what results to anticipate. There are presently just a few review publications discussing deep learning techniques for gait identification, including deep pipeline parameters. A few surveys on gait analysis have been conducted[70] [71] [72] [73] [74] [75], but the majority of survey publications concentrated on model-free gait recognition systems, ignoring model-based strategies. Despite the aforementioned benefits, gait identification real-time performance suffers because of factors like gait characteristics analyzed from various deep learning architectures and datasets. Researchers have focused on finding a method to develop a reliable gait recognition system in response to these problems [78] [79]. This survey article aims to analyze in-depth the most recent developments in gait recognition research.

*Corresponding authors
anubhaparashar1025@gmail.com (A. Parashar)

1.1. Contribution

The following are the important contributions of the paper.

1) Demonstrating the effectiveness of deep learning (DL) pipelines in resolving complex challenges in image and signal processing applications over the last decade.

2) Emphasizing the need for a comprehensive understanding of various intermediate layers and their implications in crafting robust DL pipelines.

3) Outlining the meticulous iterative refinement process involving careful selection and rigorous performance validation of each configuration.

4) Acknowledging the challenges in selecting a robust DL pipeline that can perform effectively across diverse datasets.

5) Serving as a guiding resource for researchers, facilitating a comprehensive grasp of distinct gait sensing technologies and establishing a strong foundation in deep learning concepts.

6) Providing a detailed survey of gait biometrics tailored for real-time surveillance applications, despite the relatively recent emergence of gait recognition in practical scenarios.

1.2. Organization

The paper is organized as follows. The first section provides an outline of the topics, including motivation, contribution, and organization. In Section 2, an examination of numerous factors reported in studies resulted in the development of a deep learning pipeline for gait in terms of – data collection, data types, dataset, preprocessing, deep learning techniques, feature extraction, feature reduction, regularisation, activation function, hyper-parameters, optimizer, loss function, classification, and system configuration. Section 3 discusses and highlights the benefits and shortcomings of pipeline components along with gait sensing technology. The accuracy and dataset utilized in research publications compare major features of deep learning pipelines. Section 4 is the conclusion. We list the acronyms used in this article in the abbreviations section.

2. Various Deep Learning Parameters Recognizing Gaits

Deep learning approaches successfully complete a variety of classification tasks and provide fresh perspectives on challenging covariate situations. Deep learning systems often have the drawback of behaving as a "black box" in the majority of situations; for instance, if pre-trained models are employed if hyper tuning is not performed, they seldom reveal the reasoning behind a given a choice. The deep learning pipeline for recognizing gaits is shown in Figure 3, together with all of its potential parameters. The deep learning parameters used in gait recognition are shown in Table 1-7 and Table 9. One may readily identify which deep learning pipeline can tackle certain covariate situations by looking through this table.

2.1. Dataset

In recent years, data collection technology has advanced further, and new gait identification-related training variables have been included. The gait datasets that have been utilized in the mentioned research and are currently available are examined in this section. OU-ISIR large population dataset [48] comprises subjects with two cameras surrounding the ground they are walking on. OU-ISIR multi-view large population dataset [49] is made by capturing the walking patterns of 10,307 subjects lying in the age group of 2 to 87 years. The images were captured from 14 different view angles and were 1280 by 980 pixels. Seven cameras were placed at 15-degree intervals that took images at 25fps. OU-ISIR multi-view large population dataset with pose sequence [50], is built upon OU-MVLP. Seven network cameras with a frame rate of 25 fps were used to capture the gait sequence of 10,307 subjects. The images were in RGB format and were of the dimensions 1280 by 980. In the OU-ISIR inertial sensor, dataset [50], the sensors are positioned around the waist of 744 subjects of 2-78 years of age groups. The readings were taken on a flat surface as well as an up-slope and down-slope. List of datasets used in deep learning-based gait papers are listed in table 1.

CASIA gait database A [51], consists of 12 image sequences of 20 subjects and provides four images of different view angles (0, 45, 90). There are a total of 19,139 images in the dataset. CASIA gait database B [52], provides clothing variation, carrying condition, and view angle variation. The dataset captured 11 view angles, and the dataset was built by capturing video sequences of 124 subjects. TUM gait from an audio, image, and depth database (GAID) [53], provides RGB video of 305 subjects in 3 different variations. It covers spatial-temporal covariate conditions. CMU motion of body (MoBo) database [54], provides treadmill gait of 25 subjects and covers carrying, cross-view, and speed covariates. The subjects were asked to walk at four different paces, and six cameras were used to capture the images. KY4D gait database A [55], provides images and 3D models of 42 subjects walking in a straight course and covers clothing and pose covariates. Sixteen cameras were placed along the path to capture further reconstructed images using a visual hull approach.

Dataset eGaIT [56] constitutes gait parameter validation of 101 subjects walking on a straight course. They used a wearable sensor on the lateral side of the shoes. Googlenet dataset [57] is a 22 layers deep CNN. It uses inception modules, giving the network freedom to select multiple filter sizes of convolution for every block. SZTAKI-LGA database [58] is used for activity recognition and person identification based on the gait. The dataset constitutes data from 54 subjects and ten outdoor sequences. ZJU-GaitAcc dataset [59] constitutes gait sequences of 175 subjects between the age group of 16 to 40. The dataset covers pose covariate for gait recognition. McGill dataset [60] contains raw sensor data gathered from 20 subjects that carried their phone in their pockets.

EID-M EEG dataset [61] is a sensor-based dataset with 7000 samples. These samples are collected from eight subjects with their eyes in a rest state on 14 electrodes with the help of EPOC+ at 128Hz for a duration of 54s. DVS128-Gait+ EV dataset [62] is a sensor-based dataset having 21 subjects and 100 sequences. GPJATK dataset [63] is a 3D dataset that constitutes 166 sequences of 32 subjects captured with the help of 10 mocap cameras and four video cameras. 3D flash lidar camera gait dataset [64] constitutes 16,300 people and 166 gait sequences. The Data was captured indoors and covered temporal and noisy image covariates. SisFall database [65] constitutes of 19 people and 38 gait sequences. The Data was captured indoors and covered temporal covariate. UniMiB SHAR database [66] consists of acceleration samples obtained from cellular devices of 30 subjects. ASLH database is a temporal sensor-based dataset [67] constitutes 378 gait sequences of 63 subjects captured indoors. CNU database [68] considers sensor-based covariates and constitutes 495 subjects with four gait sequences of each subject captured indoors. SIIT-CN [69] considers sensor-based covariates and contains real-time data captured indoors.

2.2. Feature Extraction

When dealing with massive amounts of raw data, feature extraction refers to the process of transforming raw data into numerical characteristics that may be analysed while retaining the information in the original data set. A large number of variables in these large data sets involve using a large number of computer resources to process them. Table 3 provides a detailed overview of the various methods used in deep learning frameworks with different feature extraction and representation techniques explored.

2.3. Feature Reduction

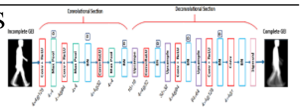
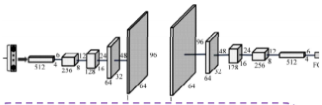
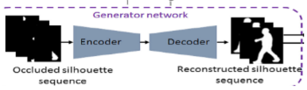
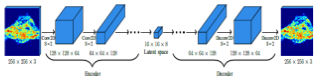
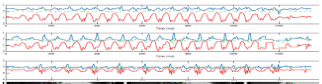
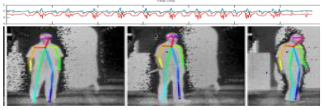
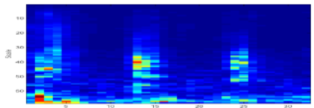
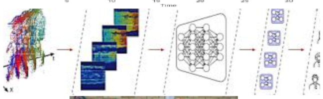
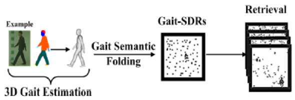
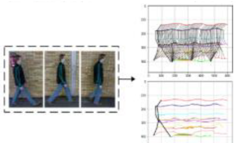
Component and discriminant analysis are two techniques that may be used to reduce dimensionality. For example, auto-encoders may transform input data into a condensed or encoded output to save on storage space or for data compression. Such datasets may be made simpler to interpret while losing the least amount of information feasible by reducing the number of dimensions using PCA. It does this by creating new variables that step-by-step reduce variance while being unrelated to one another [41] [44]. The concealed vector will be converted into the output sequence by the decoder [7] [15] [42] [44]. Average pooling is the process of figuring out the average for each patch on the map [5]. A linear transformation is a function that converts one vector space to another while maintaining each vector space's underlying structure [14]. List of feature reduction techniques used in deep learning-based gait papers are listed in table 6.

A new model-based approach for a finite element model for the 3D temporal displacement field fitting [44]. GRUs is a gating mechanism in recurrent neural networks, although they have fewer parameters than LSTMs since they don't have an output gate [4]. TensorFlow Transform is a package for pre-processing TensorFlow input data [22]. The F-score, often known as the F-measure, is a measurement of a test's accuracy. It is determined using the test's accuracy and recall [24]. The probability density function of the Gaussian distribution, which represents statistical noise, is identical to that of the normal distribution [25]. PCA may provide translation-invariant and deformation-stable transformations using the DCT (scattering transform) [29]. Batch normalization is a deep neural network training approach that standardizes the inputs to each layer of each little batch. This significantly reduces the number of training epochs necessary to develop deep networks by stabilizing the learning process and stabilizing the learning process [30]. While scattering, the Doppler Fourier Transform is a novel way of getting velocity angle distributions.

2.4. Regularization Techniques

The process of regularisation includes adding a penalty term to the error function in order to optimize a function. The extra term controls the overly fluctuating function, preventing the coefficients from reaching extreme levels. Nor-

Table 1
Feature extraction and representation techniques

Cite	Technique	Image
[1-2] [4-6] [7-14] [17] [19-23] [25-28] [30-33] [37-40] [42] [46-47]	CNN s	S 
	Encoder	
[3]	Encoder, Decoder	
[15]	Convolutional autoencoder	
[16]	Quasi-periodic signals	
[24]	3D Joint location estimator	
[29]	Scattering transform	
[36]	CNN with Residual Block	
[41]	OF, average pooling	
[43]	3D Gait Estimation	
[44]	Spatial temporal feature	
[45]	Reconstructed trajectories and encode feature embeddings	

malization is the process of meticulously dissecting tables to get rid of redundant data and undesirable characteristics like Insertion, Update, and Deletion Anomalies. It takes many steps to transform data into a tabular representation and remove redundant data from linked tables. The quantity of data provided is often what increases the effectiveness of deep learning neural networks. Data augmentation is a technique for generating new training data from existing data. Variants of the training set picture that the model is likely to see are referred to as this. List of regularization techniques used in deep learning-based gait papers are listed in table 5.

Table 2
List of parameters used in deep learning-based gait papers

Cite	Regularization techniques	Optimizer	Hyper-Parameters	Loss function
[1]	Dropout	Adam	SGD	Back-propagation
[2]	Normalization on the scores	SVM	SGD; BP	SoftMax loss function
[3]	NA	Adam	LR= 0.001	Cross-entropy loss
[4]	NA	Adam	LR=.0001 M= .9 WD= 5 * 10-4	Average loss
[5]	L2-norm	Multi-task loss	LR= 0.01 M=0.9 WD= 0.0005 Mean=0 SD= 0.01 Bias=0 SGD	Cross entropy loss
[6]	Dropout regularization	GA; PSO	LR= 3e-6	RMSprop
[7]	Gloraot initialization	Adam	LR= 0.001	Mean square error
[8]	NA	Adam	LR= 0.0005 and 0.003	Cross-entropy;
[9]	dropout = p=0.5	Back propagation	LR, M used	Mean square error
[10]	NA	NA	NA	NA
[11]	Dropout	Adam	SGD	Categorical cross entropy
[12]	NA	NA	NA	E = - sum (y' log(y))
[13]	Data augmentation	Stochastic approximation	BP; M= 0.9; LR= 0.001; WD= 5*10-4	Average loss
[14]	Dropout= 0.5	Adam	LR= 0.001	Cross-entropy loss
[15]	NA	Adam	SNE	NA
[16]	L1 norm	NA	NA	Synchronization error
[17]	NA	NA	LR= 0.0001	NA
[18]	NA	NA	NA	NA
[19]	complex co-adaptations; Dropout= 0.5	Adam	Adaptive learning rates	Back propagation
[20]	NA	Adam	SGD LR= 0.005	AdaGrad loss
[21]	Dropout= 0.6, Batch size= 32	Adam	LR= 0.0001	and RMSProp
[22]	KL divergence	Adaptive moment estimation	LR=0.0001	NA
[23]	NA	NA	NA	NA
[24]	NA	NA	NA	Triplet loss
[25]	L2 normalization	ED similarity	SGD LR= 0.01	NA
[26]	Stochastic magnitude perturbation	SGD	NA	Deep multi task
[27]	NA	NA	LR= 0.0025	Negative log-likelihood
[28]	Batch normalization	NA	NA	CCE; Contrastive triplet ranking
[29]	NA	NA	NA	Log mean square
[30]	NA	Adam	LR=LR $\times$ 0.1	NA
[31]	Dropout ratio to 0.5	Adam	NA	NA
[32]	L2Regularization; Dropout 50%	NA	LR= 0.01	Hungarian
[33]	L2 lambda - Log-uniform 10-7 to 10-3	Adam	LR= 0.0002 to 0.004	Algorithm
[34]	Batch Norm 1	RMSprop	SNE; LR= 0.001	Categorical cross-entropy
[35]	NA	NA	NA	Log-likelihood loss
[36]	NA	Adam	LR= 1e-6	NA
[37]	NA	Minimizing contour	NA	Cross entropy
[38]	NA	Adam	LR= 0.001	Least square optimization
[39]	Normalized to a fixed size	NA	NA	NA
[40]	NA	Nesterov	LR= 0.008	Softmax loss, Center loss
[41]	L2 norm	Gradient Descent	LR=0.01	Softmax loss, Center loss
[42]	NA	NA	NA	Manhattan distance
[43]	NA	NA	NA	Mean square error
[44]	Dropout layer	Adam	LR= 1 $\times$ 10-3	Softmax and Center loss
[45]	NA	Adam	LR =0.001	Contrastive loss
[46]	NA	Joint Pyramid	NA	Triplet loss
[47]	L2 regularizatio	Adam	NA	Mean squared error

Dropout is a training method in which neurons are rejected at random. They "disappeared" at random. This means that on the forward pass, their influence on downstream neuron activity is erased temporally, and any weight changes are not transferred to the neuron on the backward journey [1] [6] [9] [11] [14] [19] [21] [31] [32] [44]. Batch normalization is a deep neural network training approach that standardizes the inputs to each tiny batch. This helps to stabilize the learning process and minimizes the number of training epochs required to build deep networks [21] [28].

L2 regularisation causes the weights to be small but not zero, and it produces a non-sparse solution [32] [47]. The total of the magnitudes of the vectors in space is the L1 Norm. It is the most natural approach to measuring the distance between vectors since it is the total of the absolute differences of the vectors' components. All vector components are weighted equally in this norm [16].L2 normalization adjusts the dataset values such that the sum of the squares in each row is always greater than one. It is also known as least squares [5] [25] [41]. Data augmentation is a method of creating fresh training data from existing training data [13].

Normalization layers alter the distribution of inputs to network layers [2] [39] [81].Some neurons in complex co-adaptations are extremely reliant on others. If the independent neurons get "poor" inputs, the dependent neurons may be impacted as well, potentially affecting model performance dramatically, as may occur with overfitting [19]. The KL divergence, or Kullback-Leibler divergence, is used to calculate the difference between two probability distributions over the same variable x [22]. Gloraot initialization is the process of assigning a tiny Gaussian value with mean = 0.0 and variance depending on the weight's fan-in and fan-out [7]. The stochastic magnitude perturbation causes the process to converge to a single stationary state [26].

2.5. Activation Function

The activation function of a node in deep learning affects its output by supplying input or a group of inputs. A typical integrated circuit is a digital network of activation functions that may be switched on or off depending on the input. The nonlinear activation mechanism allows the model to build complicated mappings between the network's inputs and outputs, which is necessary for investigating and modeling complex data such as pictures, video, audio, and nonlinear or high-dimensional data sets. Rectified linear activation function (ReLU) returns 1 when the input is positive; if the input is negative, it returns 0. For example, the rectification of a linear activation function allows models to train faster and more successfully [1] [2] [4] [5] [7] [12] [13] [14] [15] [17] [19] [20] [21] [22] [26] [27] [28] [30] [31] [32] [33] [36] [38] [39] [41] [45] [47]. List of activation functions used in deep learning-based gait papers are listed in table 6.

Tanh function is derived from (-1 to 1). The benefit is that on the tanh graph, negative inputs will be substantially negative, and zero inputs will be near zero. The function may be differentiated [3] [10] [11] [18] [35] [37] [43] [47]. The Softmax function is used as the activation function in the output layer of neural network models that predict a multinomial probability distribution. Softmax activation generates a single value for each node in the output layer [3] [25] [41]. The impact of the weight estimations activation function is that extremely high densities become unusual; such estimation is often an unconscious aspect of everyday physical handling of items [16]. A radial basis function is a method of approximating multivariable functions by using linear combinations of terms based on a single univariate function. This has been radicalized so that it may be utilized in several dimensions [24]. Swish is a non-monotonic, smooth function that regularly matches or exceeds. Deep learning applies to a range of difficult fields such as image classification and machine translation. It is unbounded above and below, and the difference is due to the non-monotonic property [33].

2.6. Hyper-parameters

Hyper-parameters need expertise and plenty of trial and error. Like learning rate, batch size, momentum, and weight decay, configuring hyper-parameters is neither straightforward nor simple. List of hyper-parameters used in deep learning-based gait papers are listed in table 5. During model training, these hyper-parameters act as controls that may be adjusted. We must determine the ideal value for these hyper-parameters to receive the best outcomes from our model. For convex loss functions like Support Vector Machines and Logistic Regression, SGD is a quick and easy method for training linear classification and regression models [1] [2][5] [11] [20] [25]. A hyperparameter called the learning rate determines how much the model should change each time the model weights are updated in response to the projected errors [3] [4][5] [6] [7] [8] [9][13] [14] [17] [20] [21][25] [27] [30] [32] [33][36] [38] [40] [41] [44] [45] [82].

When the model's performance reaches a plateau, an adaptive learning rate is used to reduce the learning rate by a factor of two or an order of magnitude [19]. Find the average value of a function using the mean [5]. The word "momentum" (M=0.9) is used in the gradient descent process. Gradient descent is an optimization process that finds the direction with the steepest slope in its current state and updates it by traveling in that direction [4] [5] [9] [13]. Using an optimization approach such as gradient descent and performing backpropagation.[2] [13]. Weight decay is a regularisation approach that involves adding a modest penalty to the loss function; generally, the L2 norm of the weights [4] [5] [13]. The standard deviation of a collection of values is a measure of their variance or dispersion. A low standard deviation implies that the values are near to the set's mean, while a high standard deviation suggests that the values are dispersed across a larger range [5]. Stochastic Neighbor Embedding is a nonlinear, unsupervised approach for data exploration and visualization of high-dimensional data. SNE provides a sense of how Data is organized in a high-dimensional space [15]. The activation function may be adjusted to the left or right using the bias value to better suit the data [5]. Algorithms with low variance (but a large bias) are usually simpler. Low bias (high variance) algorithms are more difficult to implement.

2.7. Optimizer

An optimizer is a method or technique for quickly adjusting several parameters to reduce loss. Let's look at some of the most popular Deep Learning optimizers that provide respectable outcomes in reviewed gait recognition tests. List of optimizers used in deep learning-based gait papers are listed in table 5. Adam is a deep learning model training approach that uses stochastic gradient descent instead of stochastic gradient descent. Adam combines the best features of the AdaGrad and RMSProp algorithms to create an approach for optimising noisy situations with sparse gradients [1] [3] [4] [7] [8] [14] [15] [19] [20] [21] [30] [31] [33] [36] [38] [44] [45] [47]. Computing the derivative from each training data instance and calculating the update instantly is referred to as stochastic gradient descent [26]. The Nesterov Accelerated Gradient technique begins with a gradient descent step, followed by a term that resembles a momentum term but is not the same as the one found in classical momentum [41]. The Support Vector Machine returns the matching hyperplane after maximizing the geometric margin. Support vectors are the names given to such sites [2].

Stochastic approximation variable procedure with some unpredictability and uncertainty in the result [13]. Multi-task loss occurs when a shared model learns many tasks at the same time. These methods have a number of benefits, including increased data economy, decreased overfitting according to shared representations, and quick learning thanks to the use of auxiliary data [5]. A computer method known as Genetic Algorithm - PSO works by repeatedly trying to make a candidate solution better on a given quality metric [6]. Adaptive Moment Estimation is another method for estimating adaptive learning rates for each parameter (Adam). Adam maintains an exponentially decaying average of previous gradients v and an exponentially decaying average of previous squared gradients, much as Adadelta and RMSprop [22]. ED similarity metric measures the requirement [25]. Minimizing contour minimizes objective function during training with Key joints similarities [37]. Proper weight adjustment in backpropagation reduces error rates and improves model reliability by enhancing generalization. Backpropagation is a technique for doing a backward pass while modifying the parameters of a model [9].

2.8. Loss Function

A mathematical function called a loss function, often referred to as a cost function, transforms an event or the values of one or more variables into a real sum that reflects any "cost" connected to the event. In a problem of optimization, a loss function is minimized. List of loss functions used in deep learning-based gait papers are listed in table 5.

2.9. Classification

The learning exercise will provide a constrained set of categorization categories. This might be a multi-class classification with several categories or a binary classification with only two classifications (0 and 1). A mathematical function having a recognisable "S"-shaped curve, also called a sigmoid curve, is called a sigmoid function. The logistic function given by the formula S(x) = 1/1+e-x and is a popular example of a sigmoid function [6]. The Softmax classifier takes its name from the softmax function, which is used to convert raw class scores into normalised positive values that add to one, allowing the cross-entropy loss to be applied [2] [3] [5] [8] [10] [13] [14] [15] [17] [18] [20] [21] [22] [26] [27] [28] [30] [31] [32] [33] [36] [37] [38] [39] [40] [43] [44] [45] [46] [47] [83]. SVM is a type of machine learning called "supervised machine learning." It can be used to solve problems like classification and regression. It changes your data using a method called the "kernel trick," and then uses these changes to figure out the best boundary between the possible outputs [3] [12] [18] [24] [35]. A simple approach called K nearest neighbour retains all existing samples and classifies new ones based on a similarity score [25] [38] [41]. List of classification techniques used in deep learning-based gait papers are listed in table 6.

To accomplish classification, the multi-channel DCNN model integrates the learned characteristics of each channel and feeds them into a Multilayer Perceptron [1]. Support in a single class Vector Machine is an unsupervised learning method that was created for binary classification and is trained solely on 'normal' data [11]. The Radial Basis Function (RBF) is a real-valued function whose value is solely determined by the distance between the input and a fixed point

Table 3
List of parameters used in deep learning-based gait papers

Cite	Year	Feature Reduction	Activation Function	Classification	GPU/CPU
[1]	2017	NA	ReLU	Multi channels DCNN	CPU
[2]	2017	NA	Rectified linear unit	Softmax	CPU
[3]	2020	NA	Tanh; Softmax	SVM	CPU
[4]	2018	Gated Recurrent Units	Rectied Linear Unit	Softmax	CPU
[5]	2019	Average pooling	ReLU	Softmax	GPU
[6]	2019	NA	Tanh	Sigmoid activation	CPU
[7]	2019	Decoder	ReLU	Semi-supervised	CPU
[8]	2020	NA	Rectified linear unit	Softmax	GPU
[9]	2020	NA	Rectified Linear Units	Linear regression	CPU
[10]	2016	NA	Tanh	One-class Softmax	CPU
[11]	2017	NA	Tanh	One-class SVM	CPU
[12]	2017	NA	ReLU	SVM	GPU
[13]	2017	NA	ReLU	Softmax	GPU
[14]	2017	Linear transformation	ReLU	Softmax	GPU
[15]	2018	Decoder	ReLU	Softmax	GPU
[16]	2018	NA	Weight estimates	RBF network estimators	GPU
[17]	2018	NA	ReLU	Softmax	GPU
[18]	2018	NA	Tanh	Softmax; SVM classifier	CPU
[19]	2018	NA	Rectified Linear Unit	Gradient boosting	CPU
[20]	2019	NA	ReLU	Softmax	CPU
[21]	2019	NA	ReLU	Softmax	CPU
[22]	2019	TF Transform	ReLU	Softmax	CPU
[23]	2019	NA	NA	NA	CPU
[24]	2019	F- score	Radial basis function	SVM	CPU
[25]	2020	Gaussian noise; Scaling	Softmax; ReLU	k-Nearest Neighbors	GPU
[26]	2020	NA	ReLU	Log softmax	CPU
[27]	2020	NA	ReLU	Softmax	CPU
[28]	2019	NA	ReLU	Softmax	CPU
[29]	2020	DCT; Scattering transform; PCA	NA	HMM	CPU
[30]	2020	Batch normalization	ReLU	softmax	CPU
[31]	2020	Doppler Fourier Transform	ReLU	softmax	CPU
[32]	2019	NA	ReLU	Softmax	CPU
[33]	2020	NA	ReLU; PReLU LReLU; Swish	Softmax	CPU
[34]	2019	NA	ReLU	SVM	CPU
[35]	2017	NA	Tanh	SVM	CPU
[36]	2019	NA	ReLU	Softmax	GPU
[37]	2020	NA	Tanh	Softmax	CPU
[38]	2020	NA	ReLU	KNN; Softmax MLP; ANN;	CPU
[39]	2018	NA	ReLU	Softmax	CPU
[40]	2020	NA	PReLU	Softmax	CPU
[41]	2020	PCA	ReLU; Softmax	Nearest neighbour	GPU
[42]	2019	Decoder	NA	NA	CPU
[43]	2020	NA	Sigmod; Tanh	Softmax	CPU
[44]	2020	Temporal displacement	NA	Softmax	CPU
[45]	2020	Decoder; PCA	ReLU;	Softmax	CPU
[46]	2020	NA	NA	Softmax	GPU
[47]	2020	NA	Tanh; ReLU	Softmax	GPU

[16]. For regression and classification issues, gradient boosting is utilized, which results in a prediction model in the form of an ensemble of weak prediction models, often decision trees [19]. Semi-supervised learning is a hybrid of supervised and unsupervised approaches. In semi-supervised learning, an algorithm learns from a dataset that contains both labeled and unlabeled data, with the unlabeled data often being the majority [7]. When a model fails to predict the right class, log softmax penalizes it significantly [26]. In a Markov chain with a hidden Markov, the development of observable events that are dependent on internal elements that are not readily observable may be described using a model [29]. Linear regression predicts a range of probabilities from 0 to 1 [9]. The Multilayer Perceptron is a neural

network with many layers that are linked in a directed graph. This means that signals can only go in one direction across the nodes. Each node, except for the input nodes, has a nonlinear activation function [38]. The process of learning to split samples into multiple classes by discovering common characteristics between samples of known classes is known as Artificial Neural Network [38].

3. Comparative Analysis of Most Adopted Deep Learning Approach

Description of accuracy achieved in various conditions with respect to dataset used are given in table 8, 9.

The most often used datasets are shown in Figure 5 (a). CASIA B is most used gait dataset. Figure 5 (b) displays the most popular feature reduction and transformation techniques for identifying gaits. Decoders are most used for feature reduction. Figure 6 (c) depicts the most extensively used regularisation techniques in gait analysis. Dropour set to 0.5 is mostly used for regularisation. Figure 6 (d) shows the most often utilized activation functions in the deep learning pipeline for gait recognition. ReLU is the most popular activation followed by tanh and softmax. Figure 6 (e) depicts the most commonly used gait recognition hyper-parameter settings. Usually learning rate is set as 0.0001. The most often used Optimizers in gait recognition are shown in Figure 6 (f). Adam optimizer is used in most of the papers. Different loss functions used in literature are shown in Figure 7 (a). Cross entropy loss followed by softmax loss function is mostly adopted. Most adopted classification functions are shown in Figure 7 (b). In deep learning mostly softmax is used as a classifier.

4. Conclusion

While the gait recognition system is still in its early stages compared to other biometrics like fingerprint, face, voice, and iris identification, its non-intrusiveness makes it more desirable than other approaches for many applications. However, running deep learning methods in real-time has made it impossible to utilize this biometric effectively in any circumstance and is thus its limitation as applications move to the edge for privacy and security in real time gait applications. In this study, a large number of deep learning pipeline parameters for recognizing gaits were determined. We evaluated a set of datasets suitable for training deep learning pipelines to handle a range of confounders in addition to evaluating the methods' accuracy. Furthermore, we analyzed the most current deep learning models for each parameter and highlighted those that delivered remarkable results. There is a variety of possible advantages and disadvantages associated with deep learning techniques, which have been thoroughly addressed. Deep learning approaches have also been discussed in terms of potential advantages and limitations. We examined the deep learning approaches to gait identification that has been most widely used. In order to identify the gait recognition gaps that need to be addressed, comparisons of accuracy attained and datasets utilized are also summarized.

Table 4
Description of accuracy achieved in various conditions with respect to dataset used

Cite	Year	Dataset	Accuracy	Application
[1]	2017	eGaIT – embedded	80.78	Gait Recognition
[2]	2017	10 subjects	93.36, 97.06	Gait Recognition
[3]	2020	EID-M EEG	99.57	Gait Recognition
[4]	2018	ZJU-gaitAcc	98.7	gait-based authentication
[5]	2019	OU-IS	94.8	People identification
[6]	2019	CASIA B	91.3	Security and authentication
[7]	2019	Calibration dataset	21.22	Gait Generation
[8]	2020	SBHAR; UniMiB; REALDISP	97; 93.6; 94	Gait Recognition
[9]	2020	PART VI - 3D motion	96.02	Gait Recognition
[10]	2016	Raw data form ankle-IMU	80	Gait Recognition
[11]	2017	GoogleNet	82	Gait Recognition
[12]	2017	OULP	96	Gait authentication
[13]	2017	ZJU-GaitAcc	93	Person re-identification
[14]	2017	McGill dataset	96.6	Gait Recognition
		OU-IS	67.9	
[15]	2018	22 subjects Treadmill	98	Smart surveillance
[16]	2018	ZJU-GaitAcc	92.2	Gait Recognition
[17]	2018	34 participants	90	Gait Recognition
[18]	2018	McGill University	71	Gait Recognition
[19]	2018	Grayscale R-D maps	99	Gait Recognition
[20]	2019	original dataset	87.97	Gait Recognition
[21]	2019	own dataset	94.25	Identification, Authentication
[22]	2019	No dataset	94.1	Gait Recognition
[23]	2019	OU-IS	89	Gait Recognition
[24]	2019	TigerCub 3D Flash lidar camera	84.88	Gait Recognition
[25]	2020	DFNAPAS	80.2	Subject identification
		SisFall	74.58	
		UniMiB-SHAR	82	
		ASLH	81.72	
[26]	2020	CNU	96.24	Gait Recognition
		OU-IS	82.53	
[27]	2020	Identification	93.5	Gait Recognition
		Authentication	93.7	
[28]	2019	UIR	94.2	Gait Recognition
		HHAR	87.9	
[29]	2020	signals of a walking	91	Gait Recognition
[30]	2020	mmWave - radar waves	90	Gait Recognition

A. Abbreviations

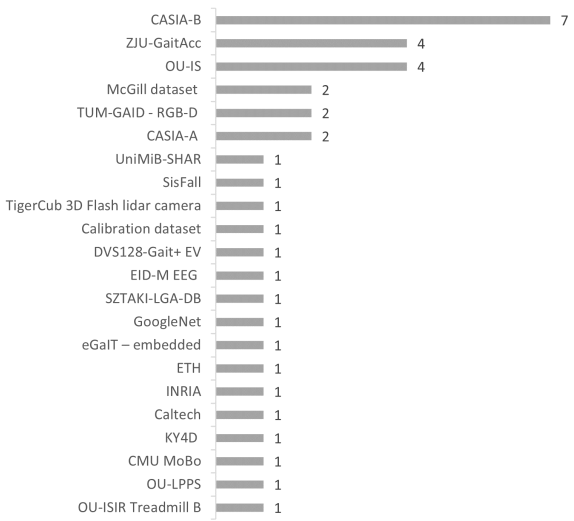
The following abbreviations are used in this manuscript:

Table 5
Description of accuracy achieved in various conditions with respect to dataset used

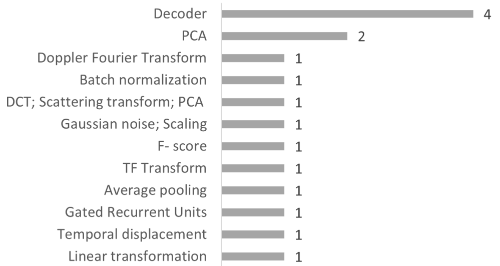
Cite	Year	Dataset	Accuracy	Application
[31]	2020	12 people data	89	Gait Recognition
[32]	2019	12 participants	84.88	Gait Recognition
[33]	2020	treadmill - pressure data	99.9	Recognition; Authorize user
[34]	2019	Floor Sensor System	93	Gait Recognition
[35]	2017	SZTAKI-LGA-DB	96.84	Gait Recognition
[36]	2019	DVS128-Gait+ EV-CASIA-B	96	Gait Recognition
[37]	2020	3D walking	97.2	Gait Recognition
[38]	2020	SIIT-CN-A, B, D, F, G	98.4	Gait Recognition
		C	100	
		E	98.9	
[39]	2018	CASIA B nm	55.91	Gait Recognition
		CASIA B BG	37.07	
		CASIA B CL	22.69	
[40]	2020	CASIA B NM	63.78	Gait Recognition
		CASIA B BG	42.5	
		CASIA B CL	31.98	
[41]	2020	CASIA B	92.95	Gait Recognition
		TUM GAID	99.78	
		OU-LPPS	98.5	
[42]	2019	CASIA A	70	Gait Recognition
[43]	2020	CMU Mobo	94	Gait Recognition
		CASIA B - Bag	91.6	
		CASIA B Coat	92.4	
		KY4D	92	
[44]	2020	CASIA A	100	Gait Recognition
		CASIA B NM	99.41	
		CASIA B BG	97.8	
		CASIA B CL	82.82	
[45]	2020	TUM GAIT	87.7	Gait Recognition
[46]	2020	CASIA B NM	73.9	Gait Recognition
		CASIA B BG	59.1	
		CASIA B CL	44.6	
[47]	2020	3D SKELETON	93.2	Gait Recognition

B. Data availability

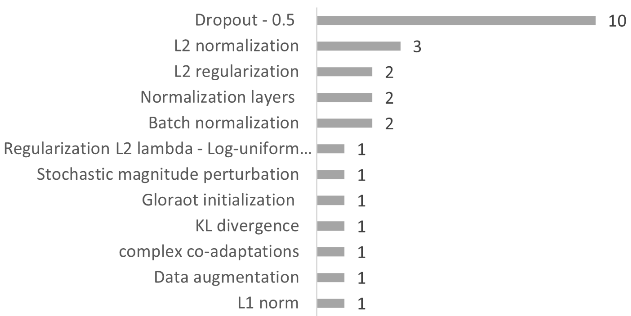
My manuscript has no associated data.



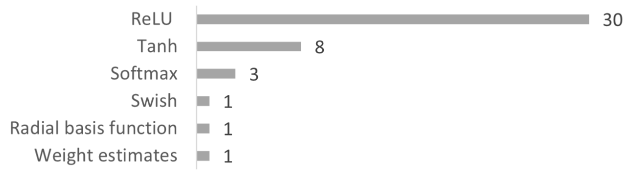
(a) Most adopted gait datasets



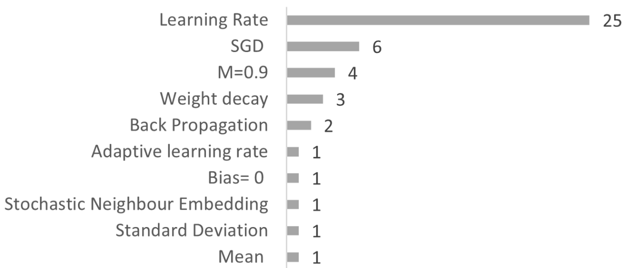
(b) Most adopted feature reduction techniques



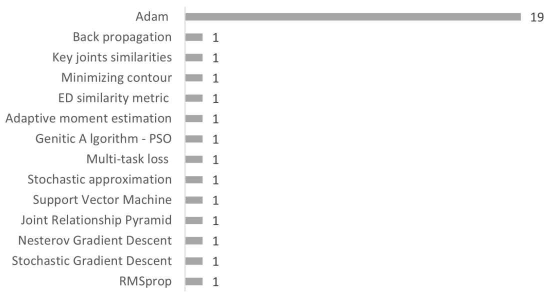
(c) Most adopted regularization techniques



(d) Most adopted activation function

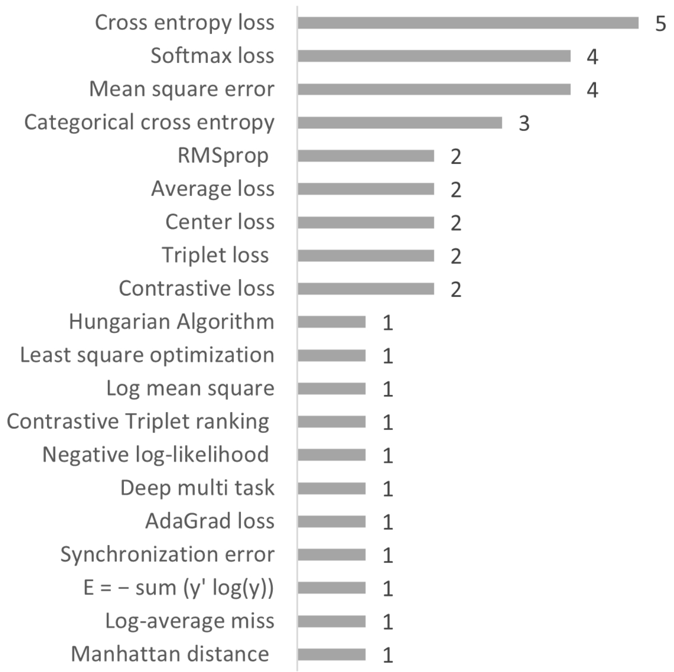


(e) Most adopted hyper-parameters

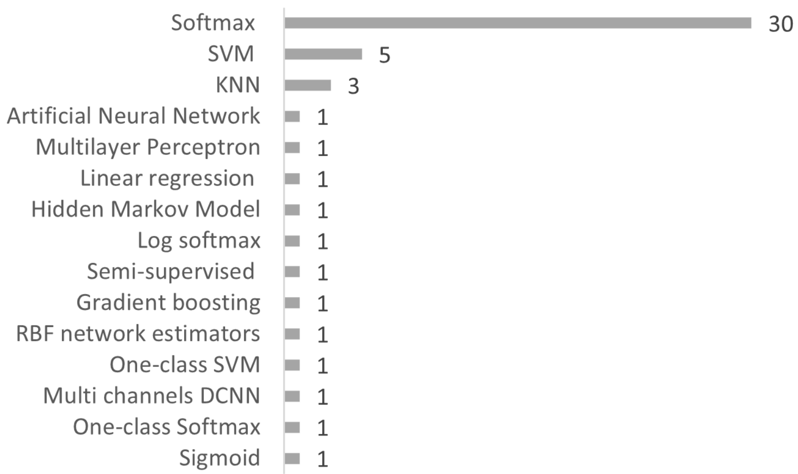


(f) Most adopted optimizer details

Figure 1: Most adopted deep learning approach



(a) Most adopted loss function details



(b) Most adopted classification functions

Figure 2: Most adopted deep learning approach

CRediT authorship contribution statement

Anubha Parashar: Conceptualization, Data acquisition, Analysis and interpretation of data, Investigation, Visualization, Writing - original draft. **Apoorva Parashar:** Data acquisition, Analysis and interpretation of data, Investigation, Visualization. **Andrea F. Abate:** Contributed to refining the ideas, Reviewing, Editing and Finalizing this paper. **Rajveer Singh Shekhawat:** Research Supervisor, Reviewing. **Imad Rida:** Contributed to refining the ideas, Reviewing and Finalizing this paper.

References

[1] J. Hannink, T. Kautz, C. F. Pasluosta, K.-G. Gasmann, J. Klucken, and B. M. Eskofier, "Sensor-Based Gait Parameter Extraction With Deep Convolutional Neural Networks," IEEE J. Biomed. Heal. Informatics, vol. 21, no. 1, pp. 85–93, Jan. 2017, doi: 10.1109/JBHI.2016.2636456.

[2] 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, Nov. 2017, doi: 10.3390/s17122735.

[3] X. Zhang, L. Yao, C. Huang, T. Gu, Z. Yang, and Y. Liu, "DeepKey: A Multimodal Biometric Authentication System via Deep Decoding Gaits and Brainwaves," ACM Trans. Intell. Syst. Technol., vol. 11, no. 4, pp. 1–24, Jul. 2020, doi: 10.1145/3393619.

[4] G. Giorgi, F. Martinelli, A. Saracino, and M. Sheikhalishahi, "Walking Through the Deep: Gait Analysis for User Authentication Through Deep Learning," in IFIP International Conference on ICT Systems Security and Privacy Protection, 2018, pp. [.5ex]–76, doi: 10.1007/978-3-319-99828-2_5.

[5] R. Delgado-Escano, F. M. Castro, J. R. Cozar, M. J. Marin-Jimenez, and N. Guil, "An End-to-End Multi-Task and Fusion CNN for Inertial-Based Gait Recognition," IEEE Access, vol. 7, pp. 1897–1908, 2019, doi: 10.1109/ACCESS.2018.2886899.

[6] P. Kumar, S. Mukherjee, R. Saini, P. Kaushik, P. P. Roy, and D. P. Dogra, "Multimodal Gait Recognition With Inertial Sensor Data and Video Using Evolutionary Algorithm," IEEE Trans. Fuzzy Syst., vol. 27, no. 5, pp. 956–965, May 2019, doi: 10.1109/TFUZZ.2018.2870590.

[7] D. Kim, M. Kim, J. Kwon, Y.-L. Park, and S. Jo, "Semi-Supervised Gait Generation With Two Microfluidic Soft Sensors," IEEE Robot. Autom. Lett., vol. 4, no. 3, pp. 2501–2507, Jul. 2019, doi: 10.1109/LRA.2019.2907431.

[8] L. Chen, Y. Zhang, and L. Peng, "METIER: A Deep Multi-Task Learning Based Activity and User Recognition Model Using Wearable Sensors," Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol., vol. 4, no. 1, pp. 1–18, Mar. 2020, doi: 10.1145/3381012.

[9] S. Davarzani et al., "Closing the Wearable Gap—Part VI: Human Gait Recognition Using Deep Learning Methodologies," Electronics, vol. 9, no. 5, p. 796, May 2020, doi: 10.3390/electronics9050796.

[10] M. Gadaleta, L. Merelli, and M. Rossi, "Human authentication from ankle motion data using convolutional neural networks," in 2016 IEEE Statistical Signal Processing Workshop (SSP), Jun. 2016, pp. 1–5, doi: 10.1109/SSP.2016.7551815.

[11] V. Prabhu and J. Whaley, "Vulnerability of deep learning-based gait biometric recognition to adversarial perturbations," CVPR 2017 CV-COPS Work., vol. 1, no. 2, 2017, [Online]. Available: https://blog.unify.id/wp-content/uploads/2017/07/Paper.pdf.

[12] K.-T. Nguyen, T.-L. Vo-Tran, D.-T. Dinh, and M.-T. Tran, "Gait Recognition with Multi-region Size Convolutional Neural Network for Authentication with Wearable Sensors," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10646 LNCS, 2017, pp. 197–212.

[13] G. Giorgi, F. Martinelli, A. Saracino, and M. Sheikhalishahi, "Try Walking in My Shoes, if You Can: Accurate Gait Recognition Through Deep Learning," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 10489 LNCS, 2017, pp. 384–395.

[14] Y. Zhao and S. Zhou, "Wearable Device-Based Gait Recognition Using Angle Embedded Gait Dynamic Images and a Convolutional Neural Network," Sensors, vol. 17, no. 3, p. 478, Feb. 2017, doi: 10.3390/s17030478.

[15] S. Abdulatif, F. Aziz, K. Armanious, B. Kleiner, B. Yang, and U. Schneider, "A Study of Human Body Characteristics Effect on Micro-Doppler-Based Person Identification using Deep Learning," Arxiv, no. November, pp. 1–6, Apr. 2018, [Online]. Available: https://ieeexplore.ieee.org/document/8835652/.

[16] W. Zeng, J. Chen, C. Yuan, F. Liu, Q. Wang, and Y. Wang, "Accelerometer-based gait recognition via deterministic learning," in 2018 Chinese Control And Decision Conference (CCDC), Jun. 2018, no. September, pp. 6280–6285, doi: 10.1109/CCDC.2018.8408232.

[17] W. Yuan and L. Zhang, "Gait Classification and Identity Authentication Using CNN," in 18th Asia Simulation Conference, AsiaSim 2018, Kyoto, Japan, 2018, vol. 946, pp. 119–128, doi: 10.1007/978-981-13-2853-4_10.

[18] M. Gadaleta and M. Rossi, "IDNet: Smartphone-based gait recognition with convolutional neural networks," Pattern Recognit., vol. 74, pp. 25–37, Feb. 2018, doi: 10.1016/j.patcog.2017.09.005.

[19] S. Abdulatif, Q. Wei, F. Aziz, B. Kleiner, and U. Schneider, "Micro-doppler based human-robot classification using ensemble and deep learning approaches," in 2018 IEEE Radar Conference (RadarConf18), Apr. 2018, pp. 1043–1048, doi: 10.1109/RADAR.2018.8378705.

[20] J. Gao, P. Gu, Q. Ren, J. Zhang, and X. Song, "Abnormal Gait Recognition Algorithm Based on LSTM-CNN Fusion Network," IEEE Access, vol. 7, pp. 163180–163190, 2019, doi: 10.1109/ACCESS.2019.2950254.

[21] D. Jung et al., "Deep Neural Network-Based Gait Classification Using Wearable Inertial Sensor Data," in 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Jul. 2019, pp. 3624–3628, doi: 10.1109/EMBC.2019.8857872.ing to Identity and Covariate Features," IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020.

[22] S. Z. Gurbuz and M. G. Amin, "Radar-Based Human-Motion Recognition With Deep Learning: Promising Applications for Indoor Monitoring," IEEE Signal Process. Mag., vol. 36, no. 4, pp. 16–28, Jul. 2019, doi: 10.1109/MSP.2018.2890128.

[23] Fernandez-Lopez, Liu-Jimenez, Kiyokawa, and Wu, “Recurrent Neural Network for Inertial Gait User Recognition in Smartphones,” Sensors, vol. 19, no. 18, p. 4054, Sep. 2019, doi: 10.3390/s19184054.

[24] N. Sadeghzadehyazdi, T. Batabyal, N. K. Dhar, B. O. Familoni, K. M. Iftekharuddin, and S. T. Acton, “GlidarCo: gait recognition by 3D skeleton estimation and biometric feature correction of flash lidar data,” pp. 1–11, May 2019, [Online]. Available: http://arxiv.org/abs/1905.07058.

[25] R. Delgado-Escaño, F. M. Castro, J. R. Cózar, M. J. Marín-Jiménez, N. Guil, and E. Casilari, “A cross-dataset deep learning-based classifier for people fall detection and identification,” Comput. Methods Programs Biomed., vol. 184, no. December, p. 105265, Feb. 2020, doi: 10.1016/j.cmpb.2019.105265.

[26] L. Tran and D. Choi, “Data Augmentation for Inertial Sensor-Based Gait Deep Neural Network,” IEEE Access, vol. 8, pp. 12364–12378, 2020, doi: 10.1109/ACCESS.2020.2966142.

[27] Q. Zou, Y. Wang, Q. Wang, Y. Zhao, and Q. Li, “Deep Learning-Based Gait Recognition Using Smartphones in the Wild,” IEEE Trans. Inf. Forensics Secur., vol. 15, pp. 1–1, 2020, doi: 10.1109/TIFS.2020.2985628.

[28] M.-C. Lee, Y. Huang, J. J.-C. Ying, C. Chen, and V. S. Tseng, “DeepIdentifier: A Deep Learning-Based Lightweight Approach for User Identity Recognition,” in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11888 LNAI, no. March, Springer International Publishing, 2019, pp. 389–405.

[29] S. Kitic, G. Puy, P. Perez, and P. Gilberton, “Scattering features for multimodal gait recognition,” in 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Nov. 2017, vol. 2018-Janua, pp. 843–847, doi: 10.1109/GlobalSIP.2017.8309079.

[30] Z. Meng et al., “Gait Recognition for Co-existing Multiple People Using Millimeter Wave Sensing,” in Proceedings of the AAAI Conference on Artificial Intelligence, 2020, no. Vol 34 No 01, [Online]. Available: https://github.com/mmGait/people-gait.

[31] P. Zhao et al., “mID: Tracking and Identifying People with Millimeter Wave Radar,” in 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS), May 2019, pp. 33–40, doi: 10.1109/DCOSS.2019.00028.

[32] A. Turner and S. Hayes, “The Classification of Minor Gait Alterations Using Wearable Sensors and Deep Learning,” IEEE Trans. Biomed. Eng., vol. 66, no. 11, pp. 3136–3145, Nov. 2019, doi: 10.1109/TBME.2019.2900863.

[33] P. Terrier, “Gait Recognition via Deep Learning of the Center-of-Pressure Trajectory,” Appl. Sci., vol. 10, no. 3, p. 774, Jan. 2020, doi: 10.3390/app10030774.

[34] O. Costilla-Reyes, R. Vera-Rodriguez, P. Scully, and K. B. Ozanyan, “Analysis of Spatio-Temporal Representations for Robust Footstep Recognition with Deep Residual Neural Networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 2, pp. 285–296, Feb. 2019, doi: 10.1109/TPAMI.2018.2799847.

[35] B. Gálai and C. Benedek, “Gait Recognition with Compact Lidar Sensors,” in Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 2017, vol. 6, pp. 426–432, doi: 10.5220/0006124404260432.

[36] Y. Wang et al., “EV-Gait: Event-Based Robust Gait Recognition Using Dynamic Vision Sensors,” in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2019, vol. 2019-June, pp. 6351–6360, doi: 10.1109/CVPR.2019.00652.

[37] J. Luo and T. Tjahjadi, “Multi-Set Canonical Correlation Analysis for 3D Abnormal Gait Behaviour Recognition Based on Virtual Sample Generation,” IEEE Access, vol. 8, pp. 32485–32501, 2020, doi: 10.1109/ACCESS.2020.2973898.

[38] P. Limcharoen, N. Khamsemanan, and C. Nattee, “View-Independent Gait Recognition Using Joint Replacement Coordinates (JRCs) and Convolutional Neural Network,” IEEE Trans. Inf. Forensics Secur., vol. 15, pp. 3430–3442, 2020, doi: 10.1109/TIFS.2020.2985535.

[39] W. An, R. Liao, S. Yu, Y. Huang, and P. C. Yuen, “Improving Gait Recognition with 3D Pose Estimation,” Int. J. Comput. Sci. Inf. Secur., vol. 1, no. April 2019, pp. 137–147, 2018, doi: 10.1007/978-3-319-97909-0_15.

[40] R. Liao, S. Yu, W. An, and Y. Huang, “A model-based gait recognition method with body pose and human prior knowledge,” Pattern Recognit., vol. 98, p. 107069, Feb. 2020, doi: 10.1016/j.patcog.2019.107069.

[41] A. Sokolova and A. Konushin, “Pose-based deep gait recognition,” IET Biometrics, vol. 8, no. 2, pp. 134–143, Mar. 2019, doi: 10.1049/iet-bmt.2018.5046.

[42] H. L. Tavares, J. B. C. Neto, J. P. Papa, D. Colombo, and A. N. Marana, “Tracking and Re-identification of People Using Soft-Biometrics,” in 2019 XV Workshop de Visão Computacional (WVC), Sep. 2019, no. March 2020, pp. 78–83, doi: 10.1109/WVC.2019.8876921.

[43] J. Luo and T. Tjahjadi, “Gait Recognition and Understanding Based on Hierarchical Temporal Memory Using 3D Gait Semantic Folding,” Sensors, vol. 20, no. 6, p. 1646, Mar. 2020, doi: 10.3390/s20061646.

[44] C. Technology, “Multi-Level Feature Fusion for Robust Pose-Based Gait Recognition using RNN,” Int. J. Comput. Sci. Inf. Secur., vol. 18, no. 1, pp. 20–31, 2020.

[45] W. Sheng and X. Li, “Siamese denoising autoencoders for joints trajectories reconstruction and robust gait recognition,” Neurocomputing, vol. 395, pp. 86–94, Jun. 2020, doi: 10.1016/j.neucom.2020.01.098.

[46] N. Li, X. Zhao, and C. Ma, “A model-based Gait Recognition Method based on Gait Graph Convolutional Networks and Joints Relationship Pyramid Mapping,” Arxiv, pp. 1–10, Apr. 2020, [Online]. Available: http://arxiv.org/abs/2005.08625.

[47] 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, doi: 10.1109/ACCESS.2020.2967845.

[48] Y. Makihara et al., “The OU-ISIR Gait Database Comprising the Treadmill Dataset,” IPSJ Trans. Comput. Vis. Appl., vol. 4, pp. 53–62, 2012, doi: 10.2197/ipsjtcva.4.53.

[49] Takemura, Noriko, et al. "Multi-view large population gait dataset and its performance evaluation for cross-view gait recognition." IPSJ Transactions on Computer Vision and Applications 10.1 (2018): 1-14"

[50] W. An et al., “Performance Evaluation of Model-based Gait on Multi-view Very Large Population Database with Pose Sequences,” IEEE Trans. Biometrics, Behav. Identity Sci., vol. XX, no. Xx, pp. 1–1, 2020, doi: 10.1109/TBIOM.2020.3008862.

[51] Liang Wang, Tieniu Tan, Huazhong Ning, and Weiming Hu, “Silhouette analysis-based gait recognition for human identification,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 25, no. 12, pp. 1505–1518, Dec. 2003, doi: 10.1109/TPAMI.2003.1251144.

[52] Shiqi Yu, Daoliang Tan, and Tieniu Tan, “A Framework for Evaluating the Effect of View Angle, Clothing and Carrying Condition on Gait Recognition,” in 18th International Conference on Pattern Recognition (ICPR’06), 2006, vol. 4, pp. 441–444, doi: 10.1109/ICPR.2006.67.

[53] M. Hofmann, J. Geiger, S. Bachmann, B. Schuller, and G. R. Institute, “The TUM gait from audio, image and depth (GAID) database,” J. Vis. Commun. Image Represent., vol. 25, no. 1, pp. 195–206, 2014.
[54] R. Gross and J. Cohn, “The CMU Motion of Body (MoBo) Dataset,” Cmu-Ri-Tr-01-18, no. June, pp. 1–11, 2001.
[55] Y. Iwashita, R. Baba, K. Ogawara, and R. Kurazume, “Person Identification from Spatio-temporal 3D Gait,” in 2010 International Conference on Emerging Security Technologies, Sep. 2010, pp. 30–35, doi: 10.1109/EST.2010.19.
[56] J. Barth et al., “Stride Segmentation during Free Walk Movements Using Multi-Dimensional Subsequence Dynamic Time Warping on Inertial Sensor Data,” Sensors, vol. 15, no. 3, pp. 6419–6440, Mar. 2015, doi: 10.3390/s150306419.
[57] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and Harnessing Adversarial Examples,” 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–11, Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.6572.
[58] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and Harnessing Adversarial Examples,” 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–11, Dec. 2014, [Online]. Available: http://arxiv.org/abs/1412.6572.
[59] Yuting Zhang, Gang Pan, Kui Jia, Minlong Lu, Yueming Wang, and Zhaohui Wu, “Accelerometer-Based Gait Recognition by Sparse Representation of Signature Points With Clusters,” IEEE Trans. Cybern., vol. 45, no. 9, pp. 1864–1875, Sep. 2015, doi: 10.1109/TCYB.2014.2361287.
[60] J. Frank, S. Mannor, J. Pineau, and D. Precup, “Time Series Analysis Using Geometric Template Matching,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 3, pp. 740–754, Mar. 2013, doi: 10.1109/TPAMI.2012.121.
[61] S. Keshishzadeh, A. Fallah, and S. Rashidi, “Improved EEG based human authentication system on large dataset,” in 2016 24th Iranian Conference on Electrical Engineering (ICEE), May 2016, pp. 1165–1169, doi: 10.1109/IranianCEE.2016.7585697.
[62] G. Gallego et al., “Event-based Vision: A Survey,” IEEE Trans. Pattern Anal. Mach. Intell., pp. 1–1, 2020, doi: 10.1109/TPAMI.2020.3008413.
[63] B. Kwolek, A. Michalczuk, T. Krzeszowski, A. Switonski, H. Josinski, and K. Wojciechowski, “Calibrated and synchronized multi-view video and motion capture dataset for evaluation of gait recognition,” Multimed. Tools Appl., vol. 78, no. 22, pp. 32437–32465, Nov. 2019, doi: 10.1007/s11042-019-07945-y.
[64] R. Horaud, M. Hansard, G. Evangelidis, and C. Ménier, “An overview of depth cameras and range scanners based on time-of-flight technologies,” Mach. Vis. Appl., vol. 27, no. 7, pp. 1005–1020, Oct. 2016, doi: 10.1007/s00138-016-0784-4.
[65] A. Sucerquia, J. López, and J. Vargas-Bonilla, “SisFall: A Fall and Movement Dataset,” Sensors, vol. 17, no. 12, p. 198, Jan. 2017, doi: 10.3390/s17010198.
[66] D. Micucci, M. Mobilio, and P. Napoletano, “UniMiB SHAR: A Dataset for Human Activity Recognition Using Acceleration Data from Smartphones,” Appl. Sci., vol. 7, no. 10, p. 1101, Oct. 2017, doi: 10.3390/app7101101.
[67] A. Özdemir, “An Analysis on Sensor Locations of the Human Body for Wearable Fall Detection Devices: Principles and Practice,” Sensors, vol. 16, no. 8, p. 1161, Jul. 2016, doi: 10.3390/s16081161.
[68] T. Hoang, D. Choi, and T. Nguyen, “On the Instability of Sensor Orientation in Gait Verification on Mobile Phone,” in Proceedings of the 12th International Conference on Security and Cryptography, 2015, pp. 148–159, doi: 10.5220/0005572001480159.
[69] N. Khamsemanan, C. Nattee, and N. Jianwattanapaisarn, “Human Identification From Freestyle Walks Using Posture-Based Gait Feature,” IEEE Trans. Inf. Forensics Secur., vol. 13, no. 1, pp. 119–128, Jan. 2018, doi: 10.1109/TIFS.2017.2738611.
[70] Connor, P., & Ross, A. (2018). Biometric recognition by gait: A survey of modalities and features. Computer vision and image understanding, 167, 1-27.
[71] Alharthi, A. S., Yunas, S. U., & Ozanyan, K. B. (2019). Deep learning for monitoring of human gait: A review. IEEE Sensors Journal, 19(21), 9575-9591.
[72] Topham, Luke K., et al. "Human Body Pose Estimation for Gait Identification: A Comprehensive Survey of Datasets and Models: A Comprehensive Survey of Datasets and Models." ACM Computing Surveys (2021).
[73] Prakash, C., Kumar, R., & Mittal, N. (2018). Recent developments in human gait research: parameters, approaches, applications, machine learning techniques, datasets and challenges. Artificial Intelligence Review, 49(1), 1-40.
[74] M. Mao and Y. Song, "Gait Recognition Based on 3D Skeleton Data and Graph Convolutional Network," 2020 IEEE International Joint Conference on Biometrics (IJCB), 2020, pp. 1-8, doi: 10.1109/IJCB48548.2020.9304916
[75] Singh, J. P., Jain, S., Arora, S., & Singh, U. P. (2021). A survey of behavioral biometric gait recognition: Current success and future perspectives. Archives of Computational Methods in Engineering, 28(1), 107-148.
[76] Masupha, L., Zuva, T., Ngwira, S., & Esan, O. (2015, December). Face recognition techniques, their advantages, disadvantages and performance evaluation. In 2015 International Conference on Computing, Communication and Security (ICCCS) (pp. 1-5). IEEE.
[77] Liu, S., & Silverman, M. (2001). A practical guide to biometric security technology. IT Professional, 3(1), 27-32.
[78] Zhu, Q., Yuan, N., Huang, J., Hao, X., & Zhang, D. (2019). Multi-modal AD classification via self-paced latent correlation analysis. Neurocomputing, 355, 143-154.
[79] Zhang, Z., Tran, L., Yin, X., Atoum, Y., Liu, X., Wan, J., & Wang, N. (2019). Gait recognition via disentangled representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4710-4719).
[80] Zou, Q., Wang, Y., Wang, Q., Zhao, Y., & Li, Q. (2020). Deep learning-based gait recognition using smartphones in the wild. IEEE Transactions on Information Forensics and Security, 15, 3197-3212.
[81] Zheng, J., Liu, X., Gu, X., Sun, Y., Gan, C., Zhang, J., ... & Yan, C. (2022). Gait Recognition in the Wild with Multi-hop Temporal Switch. arXiv preprint arXiv:2209.00355.
[82] Khan, A., Javed, M. Y., Alhaisoni, M., Tariq, U., Kadry, S., Choi, J., & Nam, Y. (2022). Human Gait Recognition Using Deep Learning and Improved Ant Colony Optimization. Comput. Mater. Cont, 70, 2113-2130.
[83] Giorgi, G., Saracino, A., & Martinelli, F. (2021). Using recurrent neural networks for continuous authentication through gait analysis. Pattern Recognition Letters, 147, 157-163.
[84] Saber, S., Amin, K. M., & Adel Hammad, M. (2021). An Efficient Person Re-Identification Method Based on Deep Transfer Learning Techniques. IJCI. International Journal of Computers and Information, 8(2), 94-99.

Anubha Parashar is a Ph.D. candidate at Manipal University Jaipur, India. She received her M.Tech degree in Computer Science and Engineering from VCE Rohtak, India, in 2016 and her B.Tech degree in Computer Science and Engineering from PDMCE Bahadurgarh, India, in 2013. Her research interests include Gait Recognition, Biometrics, Deep Learning, Computer Vision, Pattern Recognition, and IoT.

Apoorva Parashar is AI Consultant in Emerging technology, Mahindra Integrated Business Solutions, Mumbai. She has done her M.Tech and B.Tech in Computer Science and Engineering from Maharshi Dayanand University, Rohtak, India. She provides solutions in computer vision for the real-time processing of video analysis. Her research interests include Computer Vision, Deep Learning, Medical Imaging, Biometrics, IoT, and Artificial Intelligence.

Dr. Abate is a member of the IEEE Haptics Technical Committee and a member of the the International Association for Pattern Recognition. His current research interests include multibiometric systems, virtual/augmented/mixed reality, haptics and human–computer interaction. He has authored many scientific papers published in scientific journals and proceedings of refereed international conferences and co-edited one book. He currently serves as Associate Editor for Pattern Recognition Letters and IEEE Access.

Imad Rida received the Ph.D. degree in computer science from Normandy University, MontSaint-Aignan, France, in 2017. He is currently an Associate Professor with the University of Technology of Compiègne, Compiègne, France. His research interests include machine learning, pattern recognition, and signal/ image processing.