TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 4, August 2020, pp. 1954~1961
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i4.15200

Strategy to determine the foot plantar center of pressure of a person through deep learning neural networks

Henry Hernández Martínez[1], Holman Montiel Ariza[2], Luz Andrea Gaviria R.[3]
[1,2]Facultad Tecnológica, Universidad Distrital Francisco José de Caldas, Colombia
[3]Universidad Internacional de la Rioja, Spain

Article Info

Article history:

Received Jan 8, 2020
Revised Mar 22, 2020
Accepted Mar 12, 2020

Keywords:

Artificial neural network
Classifier
Computational learning method
Deep learning

ABSTRACT

Some case studies treated by physiotherapists or orthopedists to measure the alignment of the lower extremities during a gait cycle are based on empirical methods of visual observation. This methodology does not guarantee total success, since it depends on the experience of the specialist, what can cause irreversible damage to patients, such as: hip displacement, wear and overload of the joints of a single lower limb. Although, this problem has been addressed in the investigation by means of devices implementation with sensors or methods of processing sequences of images and videos, this topic is still under investigation because the current methods depend on many external elements and data given by an expert in the area. Therefore, this paper proposes a partial solution to this problem by systematizing the experience of a specialist through a computational learning method.

Corresponding Author:

Holman Montiel Ariza
Facultad Tecnológica,
Universidad Distrital Francisco José de Caldas,
Cll 68 D Bis A Sur No. 49F – 70, Bogotá D.C, Colombia.
Email: hmontiela@udistrital.edu.co

1. INTRODUCTION

Gait analysis are techniques that are based on the observation of continuous patterns of movement. These patterns are compared with a movement pattern considered normal to determine possible pathologies. Normally, these movement patterns are known as gait cycles and have three different phases. The first phase is support and consists of supporting the heel on the ground, the second is when the person is about to take off the toes of the floor and, finally, the third phase is when the person lifts the foot of the ground, supports the other foot and starts the cycle again [1-3]. The gait analysis is complemented by a stability analysis in a person's bi-pedestal position, which allows determining the plantar center of pressure center (COP) and the patient's center of mass (COM). The objective of this analysis is to determine the change produced by the weight and morphology of the lower extremities, on the body balance, in order to increase the amount of available data obtained during a gait analysis [4, 5].

Although, the ways of keeping track of the results during gait analysis are done manually, to reduce the margin of error, techniques have been developed that systematize this process [6]. Some of them are based on software and implement physical sensors or processing of images and videos [7]. In the first case, a group of sensors is connected to a device that is normally a computer and an attempt is made to re-construct a signal from measurements. The results obtained are compared against a recorded signal from a patient with parameters considered normal or control and the deviation of the obtained signal regarding the control signal is analyzed to determine a possible pathology [8]. Analogously, in the analysis based on image and video processing, a signal associated with the movement of the subject is reconstructed from the capture of a sequence of images and identification of regions of interest, and in the same way it is compared with other data of control [9].

Unlike these techniques, the studies of the COP and COM are mostly not performed with image processing techniques, because they do not have a resolution that allows to measure the angles of deviation of the extremities. Therefore, techniques based on sensors are implemented, such as the balance platforms, pressure sensors or scales and scales. In some of the mentioned cases there is a specialized software, which presents the regions of the feet and the effect of body posture on them through a color map in a graphical user interface [10, 11]. Gait analysis is a widely used tool during different medical examinations, since it allows medical specialists, orthopedists or physiotherapists to determine several groups of pathologies based on the variations that the walking pattern of a patient treated against a normal one. However, this type of analysis is usually performed empirically and depends on the experience of the person when evaluating the results to issue an opinion. This means that, depending on the observations made by a specialist, the results are subject to multiple interpretations given the variability of concepts that exists between specialists to treat a patient [12].

This topic has many references within the state of the art due to the large number of specialists working in the area [13-15]. Therefore, this paper proposes a partial solution to the problem of obtaining and reproducibility of the results of such analysis, by systematizing the obtaining of POP behavior through a bio-inspired computational learning technique, since these techniques are relatively simple to implement and have a low computational cost compared to other mathematical or analytical techniques or models [15, 16]. Recapitulating, the proposed technique incorporates a deep learning neuronal network, which was trained with real patient data to validate their behavior from data measured using sensors. This contribution is described in detail in the following sections, which are organized as follows; sections 2 and 3 present the general definitions and methodology used respectively, and section 4 presents the results obtained.

2. RESEARCH METHOD

The contribution made is based on the joint operation of a WII Balance Board platform, an application developed entirely in PYTHON and an artificial neural network (ANN). Next, a description of each of the elements used is presented.

2.1. WII balance board platform

A plantar pressure platform is an element that allows a specialist to measure the plantar pressure center (COP) to estimate the balance of the lower extremities of the human body. In this case, a Nintendo WII balance board (NBB) platform was used as emulator for this type of platform and thus to determine the center of gravity of a group of users from the pressure exerted by the plantar region of the lower extremities [17, 18]. The NBB has four strain gauges (one at each end), a control circuit that allows it to link with other devices that implement the L2CAP protocol to transfer and receive information, this protocol that allows the sending of data packets directly by the link manager without the need to connect to a server, see Figure 1 [18, 19].

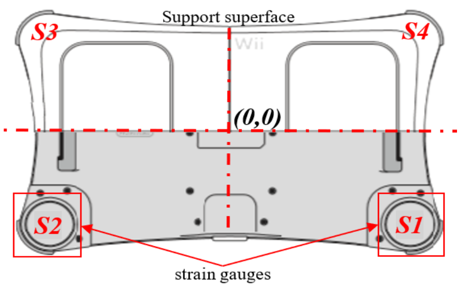


Figure 1. Platform Nintendo WII balance board (NBB)

This protocol is versatile and establishes a point-to-point network between two devices that allows information through data packets, which are coded (the packet is divided into several smaller packets) into the transmitter and decoded into the receiver (the packages are joined together to form the original). In this case, the NBB forms packets of 32-bit values and before sending them, segments them into 8-bit data groups and the computer assembles the packet again. This segmentation allows the protocol to have a relatively high level of quality of service (QoS) compared to other protocols, since it facilitates the transit of large volumes of information by dividing a packet into small segments.

The NBB requires three phases prior to its operation to start in continuous operation mode. In the first phase, the computer must establish a point-to-point network with the NBB activating the L2CAP protocol and indicating to the Bluetooth terminal that the physical address of the NBB is the link device. In the second phase, the computer sends two sequences of integers to set a level of zero in each sensor and thus calibrate and adjust the NBB, considering that the elements that are on it must be removed. In the third phase, the computer sends a start bit to confirm the completion of the calibration routine and a command that initiates the continuous sending of packages with the status information of the strain gauges or sensors (see Algorithm 1).

When initiating the process of continuous sending of the sensors' information, the computer can estimate a value of COP using some mathematical transformations following a protocol. On the one hand, the developed technique decodes the information of the sensors (1) and (2) to determine the weight in kg detected by each gauge (x [n]). It should be added that the average of 100 data (Gn) is taken as an effective measure of each sensor, that is, the sum of the average of each gauge (Pt) is the total weight of the object that is on the platform, illustrated in (2).

$$G_n[n] = \frac{\sum_{n=1}^{99} x[n]+x[n-1]}{100} \quad (1)$$

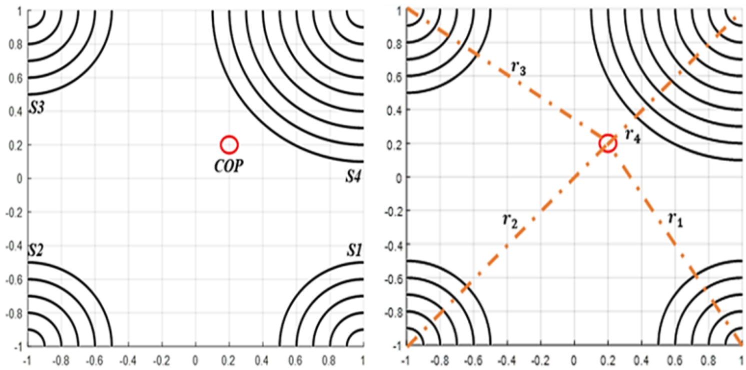
$$P_t = \sum_{n=1}^{4} \overline{G_n} \quad (2)$$

On the other, a technique based on the law of universal gravitation was used to estimate the COP. This technique proposes to increase the load on a sensor that increases its attraction force on a particle (COP), therefore, having 4 sensors each one exerts a different attraction force on the particle [20]. When the sensors stabilize with a constant load the particle will find a relative fixed position. In this case, the force of attraction (3) resulting (Fn) is estimated taking into account that the mass of the particle (m_1) is smaller than the mass of any sensor (m_n), the mass of each sensor expands according to the load present, see Figure 2 (a) and the distance between the masses (r_n) is the magnitude between the centers of mass, as show in Figure 2 (b) [21].

$$F_n = \frac{G(m_1 * m_n)}{r_n^2} \quad (3)$$

Algorithm 1. Initiation of the NBB

```
Start Variables;
Match devices using L2CAP ();
Start calibration of the NBB without load ();
If Calibration completed then
   Activate continuous operation of the NBB ();
While Bit of continuous operation = True do
      X [n] = Decode received information ();
      Gt [n] = Average (X [n]);
      Pt = Add (Gt);
End While
End If
```



(a) (b)

Figure 2. Representation of the particles and their masses;
(a) representation of the COP, (b) distance between the masses

2.2. Deep learning neuronal network

Deep learning neural networks (DLNN) systematize large volumes of information by organizing complex data structures that are not achieved with an artificial neural network (ANN). In addition, DLNNs are versatile enough to be implemented in computers without high performance, because data structures are represented by quantitative mathematical models [15, 22]. The proposed application implements the DLNN architecture of the TENSORFLOW library of the PYTHON programming language. This architecture is like ANN, since input signals propagate from input to output through layers consisting of weights, thresholds, and multiple mathematical transformations [16, 23, 24]. An ANN in its simplest form (Figure 3 (a)) consists of; an input layer, an output layer and a hidden layer, which allows you to modify the weights of the hidden layer according to the input values and the margin of error obtained at the output. Unlike ANN, DLNN (Figure 3 (b)) has more than two hidden layers and many cascaded neurons to perform some transformations of the training data from the input layer to the output layer [16, 25, 26].

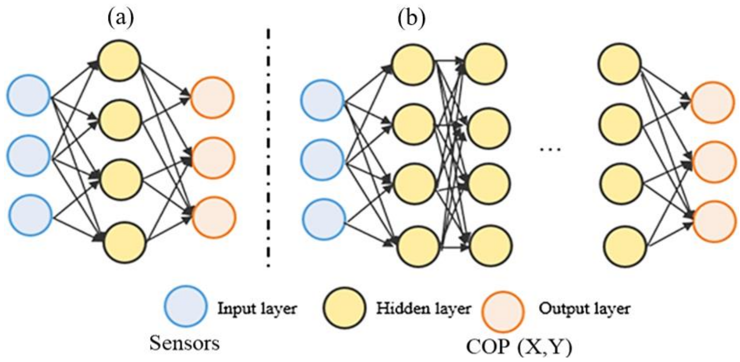


Figure 3. Topology of two neural networks; (a) artificial neural network, (b) neural network of deep learning

The DLNN can be trained with different algorithms, whose main function is to estimate the weights of the neurons in the different layers of the network. Initially, the training algorithm assigns weights to each neuron with stochastic values, zeroes or ones, which is previously defined by the user. Then, an optimization algorithm estimates the output value of the network to adjust the weights from the error between the data obtained and the training data. Finally, if the execution of the training algorithm is not successful, that is, the margin of error between the training data and the data obtained does not converge to zero, the number of training cycles must be increased, or the training algorithm must be redefined.

The configuration process of the DLNN parameters is manual, since there is no methodology for the selection of the parameters from the training data. Therefore, there is a great diversity of training algorithms, activation functions and ways to configure a DLNN. Normally, a training algorithm is a strategy that adjusts the weights of the network until finding a configuration that reduces the margin of error between the training data and the output data to approximately zero, for example, the ADAM algorithm that transforms the weights of network in a matrix to simplify them through interpolation and minimization of the factors using the descending gradient. However, the output values of each network are restricted to the limits established by the activation functions. These functions establish the way in which the response of each neuron changes and its model has a conventional form like that of mathematical functions, step, exponential or Gaussian [21]. The following section will structurally show the configurations used for the DLNN (number of neurons, optimization algorithms among other aspects), additionally the proposed algorithm for information processing will be described in detail.

3. DEVELOPMENT AND IMPLEMENTATION

The developed technique implements a neural network of deep learning (DLNN), which estimates the position of the COP using the sensor readings as input data. That is, the data from the sensors are the predictors and the position of the COP is the objective function. When grouping these data an array of 10,000 instances with 6 attributes was built and the DLNN trained with 80% and validated with 20% of the available instances, illustrated in Table 1.

Table 1. Segment of the instances used in the DLNN training

Strain Gage [kg] (Sensor Inputs)				COP [cm] (Output)	
S1	S2	S3	S4	Position X	Position Y
20.00	20.00	20.00	20.00	0.00	0.00
25.00	15.00	10.00	10.00	-2.23	1.65
10.23	10.30	5.00	20.00	3.49	-2.34
0.00	0.00	0.00	25.00	-10.00	-10.00
0.00	0.00	25.00	0.00	10.00	-10.00
4.00	2.00	1.65	3.60	0.65	1.23

The selection of the training and validation instances was stochastic. After completing the selection of the instances, their attributes were normalized, to reduce the search space of the objective function and to start the training process. Finally, several configurations of the DLNN were modified as the number of neurons, activation functions, training algorithms, among other parameters, presented on Table 2. The trained DLNN is exported as a class to be used in the developed application. This application and the training of the DLNN were carried out in a computer with operating system UBUNTU 18.04, an Intel inside TM core i3 processor, 8 GB of RAM, a hard drive of 240 GB and a Bluetooth v3.0 system integrated in a WIFI card of brand intel. The operation of the application is described in detail in Algorithm 2, where the process is observed from the reception of the frame given by the NBB to the interpretation made by the DLNN.

Table 2. Three DLNN configurations implemented to determine the COP

Description	Configuration DLNN 1	DLNN 2	DLNN 3
Quantity of neurons	1000	1000	1000
Activation function	Hyperbolic tangent	Sigmoid	Exponential
Algorithm to assign initial values to weights	Zeros	Random with uniform distribution	Ones
Number of hidden layers	100	100	100
Optimizing algorithm	Nadam	Adam	Adagrad
Strategy to estimate the error	Error absolute medium square	Error medium square	Hinge

Algorithm 2. User application

```
Function DLNN (Error margin)
    Start variables ()
    Threshold = 1 * 10 ^ -9
    Start DLNN ()
    Start weights of the DLNN ()
    Assign activation function to the DLNN ()
Expand hidden layers of the DLNN up to the defined amount ()
     While Threshold less than the Error margin do
          Optimize the weights of the DLNN ()
          Estimate the margin of error
     End While
     Export DLNN class ()
     Run GUI ()
End DLNN

Function GUI ()
Load libraries TKINTER, TENSORFLOW, MATPLOTLIB, MATH
Load DLNN
Start variables
Start BLUETOOTH port
Create reception thread from communication port
Create sending thread from the communication port
Start graphical interface objects
Start NBB
While the graphic interface is active, do
       Check reception
       If reception is full then
              Decompose reception into three parts and save it in buffer
              If buffer [0] == 0 then
                    Battery status = buffer [1]
                    Pushbutton state = buffer [2]
                    Empty buffer
```

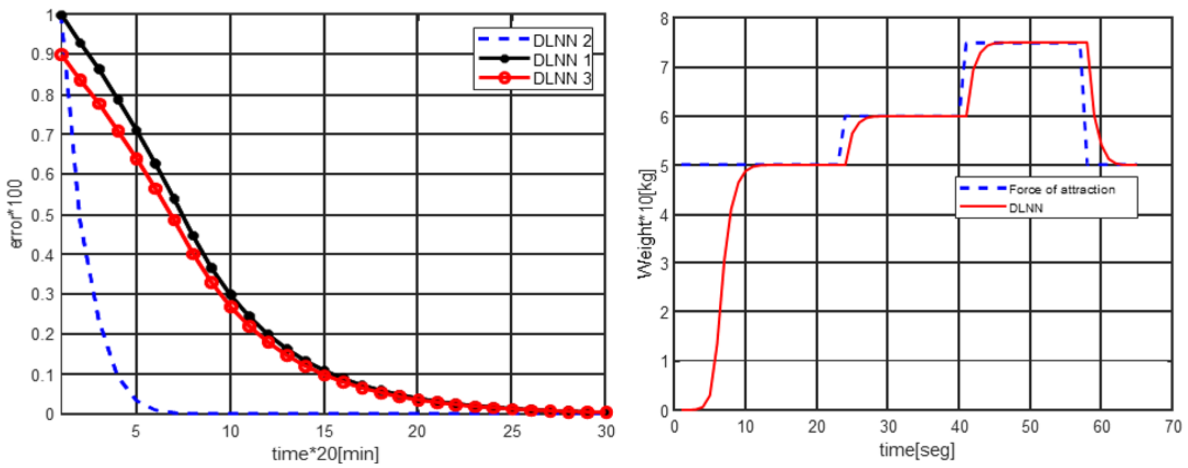
```
                Else if buffer [0] == 1 then
                     Wait for the calibration routine to finish
                     Empty buffer
                Else if buffer [0] == 2 then
                     S1 = buffer [1] << 8
                     S2 = buffer [1]
                     S3 = buffer [2] << 8
                     S4 = buffer [2]
                     Convert S1, S2, S3, S4 in kilograms
                     Empty buffer
                 Else
                     Empty buffer
                 End if
        End if
           S1, S2, S3, S4 = Normalize (S1, S2, S3, S4)
           [X, Y] = DLNN (S1, S2, S3, S4)
           Graph (S1, S2, S3, S4, X, Y)

End While
```

4. RESULTS AND ANALYSIS

The DLNN was trained with the three configurations indicated in Table 2 and the records mentioned in the previous section (see Table 1). As a result, the trend graphs are presented in Figure 4 (a). which relate the training time and the margin of error of the DLNN in each configuration. Based on this, the configuration number two of the DLNN was selected for the development of the application, since it converged more quickly than the other two. When developing the application, the DLNN response was compared with the results obtained when implementing the force of attraction method, which was carried out in two ways:

– The COP of five different people was fixed in the coordinate (0, 0) and it was verified that the sum of the charges in each gauge corresponds to the real weight reported by a scale are shown on Table 3.
– Then a load change was made on the platform in real time and the response of both techniques was graphically compared. It is illustrated in Figure 4 (b).

As can be seen, the attraction force method is highly reactive, i.e. it reflects the user's weight in short times (< 1 second). Therefore, this method was used as a validation parameter to establish which learning technique replicates sensor information appropriately (Table 4). This validation was carried out through the Wilcoxon test, in which 1000 samples measured during the operation of the application were used to implement each strategy by modifying the weight on the platform as in the dotted line of Figure 4 (b). Finally, with the error estimated with each technique, a diagram of boxes and whiskers was constructed, as show in Figure 5.



(a) (b)

Figure 4. Behavior of the Nintendo balance board; (a) behavior of DLNN with 3 different configurations, (b) behavior of the proposed technique against the method of attraction force

Table 3. Results obtained by modifying the load on the NBB

Reported weight [kg]			Margin of error [%]	
Bascule	Force of attraction	DLNN	Force of attraction	DLNN
55	54.9	55.5	0.18	0.90
65	66.1	65.2	1.69	0.30
70	70.9	69.7	1.28	0.42
78	78.4	77.7	0.51	0.38
98	95.2	97.3	2.85	0.71
120	----	120	----	0.00

Table 4. Results obtained by comparing the results obtained with other methods and the force of attraction using the Wilcoxon test

Combination	p
Attractive force - DLNN 1	0.88
Attractive force - DLNN 2	0.15
Attractive force - DLNN 3	0.10

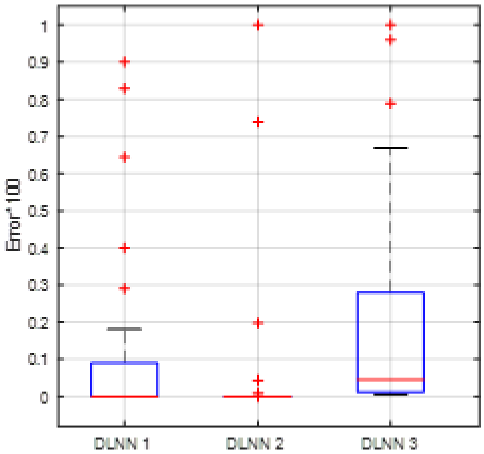


Figure 5. Box and whisker diagrams of assessed learning strategies

5. CONCLUSIONS

As can be seen in the graph in Figure 4 (a), DLNN performance improves significantly when the initial weights are configured stochastically and not with constant numbers as in the other two cases. Moreover, in this case, in configuration two, the algorithm of weight optimization and the way of error margin correction reduces the DLNN convergence time, which is advantageous because it reduces the computational time in the DLNN training time. As shown in Table 3 with the development of this application it was proved that the attraction force method does not predict COP for people with weights greater than 100 kg. This gives another advantage to the DLNN, since the proposed model allows to validate the COP of people with a weight of up to 120 kg. This weight is the maximum allowed by the WBB, because each strain gauge detect loads up to 30 kg.

Finally, as can be seen in the graph in Figure 4 (b), DLNN has a waiting time to stabilize before predicting the COP value, which is not the case with the technique based on the law of attraction. This is because the law of attraction is based on a mathematical transformation whose model is much simpler to implement than that of DLNN. However, DLNN predicts values with a higher resolution, i.e. it can predict with three decimals of accuracy and the other method with only two. In addition, as shown in Table 4 and Figure 5, DLNN 2 presented a lower variance, a high level of coincidence versus the attraction force method, a margin of error close to zero, and an accurate response to sensor information versus the other techniques tested.

REFERENCES

[1] H. Ma and W. Liao, "Human Gait Modeling and Analysis Using a Semi-Markov Process With Ground Reaction Forces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 6, pp. 597-607, June 2017.

[2] R. G. Birdal, A. Sertbaş and B. Mïhendisliği, "Human Identification Based on Gait Analysis: A survey," *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, Sarajevo, pp. 489-493, 2018.
[3] D. Tao, X. Li, X. Wu and S. J. Maybank, "General Tensor Discriminant Analysis and Gabor Features for Gait Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1700-1715, October 2007.
[4] P. Vanitchatchavan, "Termination of human gait," *2009 IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, TX, pp. 3169-3174, 2009.
[5] K. Arai and R. Andrie, "Gait Recognition Method Based on Wavelet Transformation and its Evaluation with Chinese Academy of Sciences (CASIA) Gait Database as a Human Gait Recognition Dataset," *2012 Ninth International Conference on Information Technology - New Generations*, Las Vegas, NV, pp. 656-661, 2012.
[6] D. I. Stoia and M. Toth-Tascau, "Influence of treadmill velocity on joint angles of lower limbs during human gait," *2011 E-Health and Bioengineering Conference (EHB)*, pp. 1-4, 2011.
[7] A. Vieira et al., "Software for human gait analysis and classification," *2015 IEEE 4th Portuguese Meeting on Bioengineering (ENBENG)*, Porto, pp. 1-1, 2015.
[8] R. Martín-Félez, J. Ortells and R. A. Mollineda, "Exploring the effects of video length on gait recognition", *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Tsukuba, pp. 3411-3414, 2012.
[9] K. Gui, H. Liu and D. Zhang, "Toward Multimodal Human–Robot Interaction to Enhance Active Participation of Users in Gait Rehabilitation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 2054-2066, Nov 2017.
[10] J. Hu and K. Sun, "Human gait estimation using a reduced number of accelerometers," *Proceedings of SICE Annual Conference* 2010, Taipei, pp. 1905-1909, 2010.
[11] H. Sobral et al., "Human gait analysis using instrumented shoes," *2015 IEEE 4th Portuguese Meeting on Bioengineering (ENBENG)*, Porto, pp. 1-1, 2015.
[12] S. Jung and M. S. Nixon, "Estimation of 3D head region using gait motion for surveillance video," *4th International Conference on Imaging for Crime Detection and Prevention 2011 (ICDP 2011)*, London, pp. 1-6, 2011.
[13] M. Qi, "Gait based human identification in surveillance videos," *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, Guilin, pp. 2317-2322, 2017.
[14] J. Lu and Y. Tan, "View recognition of human gait sequences in videos," *2010 IEEE International Conference on Image Processing*, Hong Kong, pp. 2457-2460, 2010.
[15] T. K. Bajwa, S. Garg and K. Saurabh, "GAIT analysis for identification by using SVM with K-NN and NN techniques," *2016 Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, Waknaghat, pp. 259-263, 2016.
[16] A. Sevik, P. Erdogmus and E. Yalein, "Font and Turkish Letter Recognition in Images with Deep Learning", *2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT)*, Ankara, Turkey, pp. 61-64, 2018.
[17] J. E. Deutsch, D. Robbins, J. Morrison and P. Guarrera Bowlby, "Wii-based compared to standard of care balance and mobility rehabilitation for two individual post-stroke", *2009 Virtual Rehabilitation International Conference*, Haifa, pp. 117-120, 2009.
[18] G. D'Addio, L. Iuppariello, F. Gallo, P. Bifulco, M. Cesarelli and B. Lanzillo, "Comparison between clinical and instrumental assessing using Wii Fit system on balance control," *2014 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, Lisboa, pp. 1-5, 2014.
[19] Y. Hua and Y. Zou, "Analysis of the packet transferring in L2CAP layer of Bluetooth v2.x+EDR," *2008 International Conference on Information and Automation*, Changsha, pp. 753-758, 2008.
[20] Torres Zambrano Jenny, Pérez Julián, "Diseño de una herramienta computacional para estimar el COP de una persona a través de una plataforma WII Balance Board," *Repositorio Institucional Universidad Distrital - RIUD*, Bogotá, 2019.
[21] J. Li and N. Dong, "Gravitational Search Algorithm with a New Technique," *2017 13th International Conference on Computational Intelligence and Security (CIS)*, Hong Kong, pp. 516-519, 2017. doi: 10.1109/CIS.2017.00120.
[22] D. Zhang, X. Han and C. Deng, "Review on the research and practice of deep learning and reinforcement learning in smart grids," *CSEE Journal of Power and Energy Systems*, vol. 4, no. 3, pp. 362-370, September 2018.
[23] G. P. Zhang, "Neural networks for classification: a survey", *IEEE Transactions on Systems Man & Cybernetics Part C Applications & Reviews*, vol. 30, no. 4, pp. 451-462, 2000.
[24] M. A. Abu et al., "A study on Image Classification based on Deep Learning and TensorFlow," *International Journal of Engineering Research and Technology*, vol. 12, no. 4, pp. 563-569, 2019.
[25] C. Guojin, Z. Miaofen, Y. Honghao and Li Yan, "Application of Neural Networks in Image Definition Recognition," *2007 IEEE International Conference on Signal Processing and Communications*, Dubai, pp. 1207-1210, 2007.
[26] Sugiarti et al., "An Artificial Neural Network Approach for Detecting Skin Cancer," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 2, pp. 788-793, 2019.