MJPAS Vol. 1, No. 1 (2023) 117-132

Mustansiriyah Journal of Pure and Applied Sciences
https://mjpas.uomustansiriyah.edu.iq
Print ISSN: 2957-9910 Online ISSN: 2957-9929

People Recognition Based on Gait and Neural Network

Razan E. Al-Bayati[1] and Ziad M. Abood[2]

[1]**Department of Computer Science, College of Education, Mustansiriyah University,**

[2]**Department of Physics, College of Education, Mustansiriyah University,**

Baghdad, Iraq ([2] dr.ziadmabood@uomustansiriyah.edu.iq)

A B S T R A C T

Walking is one in every of life science that helps to see the identity of an individual while not his data and from an excellent distance during. The main objective of this study is to design cheap gait identification system by using digital camera and laptop. In this study, the proposed system consists of main stages gaiting video, 3D key joint position for each frame determine, feature extraction, cleaning features, and identification. In the first phase, Gaiting video human body can be identified using a digital camera, Secondly, 3D key joint position for each frame determine. Thread, the features extraction phase, two methods have been used to extract features dynamics gait feature. Four, cleaning features using linear discriminant analysis. Finally, Identification by using dense neural network classifiers have been applied set of the feature. The proposed system was tested on our dataset consisting of 126 gait videos of 27 people, where the results after using 80% of the data for training and 20% for the testing showed achieving accuracy, the results of the dense classifier were for the dynamics gait feature 80%.

Keywords: *Gait Recognition, Dynamic Gait Features, Dense Neural Network, Angle, Gait Cycle.*

1. Introduction

Gait recognition is a subset of biometrics that has the benefit (over other biometrics) of being unobtrusive since it does not use body-invasive sensors to acquire gait data. Gait recognition is an appealing surveillance modality since it may be conducted at a distance and covertly. People from disadvantaged areas have gait patterns that indicate the existence of identification information [1]. Gait recognition may be performed using low-resolution movies and minimal instruments. Individuals' cooperation is not required for gait recognition. When additional characteristics, such as faces and fingerprints, are obscured, gait recognition can perform effectively. Gait characteristics are notoriously difficult to

*Corresponding author: Ziad. M. Abood
E-mail address: dr.ziadmabood@uomustansiriyah.edu.iq

imitate [2], Two major types of studies have been undertaken to identify an individual by evaluating gait patterns: Model-free approaches they construct gait signatures directly from silhouettes recovered from video sequences using background removal [3] and model-based approaches use the human body's structure and mobility as a model, extracting characteristics from bodily components such as the foot, knee, ankle, hip, wrist, shoulder, torso, thigh, head, and hand [4].

There are two type of gait feature; astatic characteristic is one that does not vary much during the walking process, such as height, the length of skeletons, including the length of legs and arms. To some extent, the individual may be recognized based on static bodily characteristics. Because of the symmetry of the human body, the length of the limbs on both sides is normally viewed as equal [5]. The dynamic characteristics change at any point during the walking process, and the change always has a periodicity [6]. Given many researches the angles of swing limbs during walking are remarkable dynamic gait features. For this reason, four groups of swing angles of upper limbs, arm and forearm, and lower limbs, thigh and crus, are defined as shown in Figure (1), and denoted as a1… a8. Here, a2 is taken as the example for illustration. The coordinate at knee right is denoted as (x, y, z), and coordinate at ankel_right is denoted as (x, y, z), so a2 Can be calculated as [39]:

$$\tan\angle a_1 = \left(\frac{x - x_1}{y - y_1}\right) \quad \Rightarrow \quad a_1 = \arctan\left(\frac{x - x_1}{y - y_1}\right) \tag{1}$$

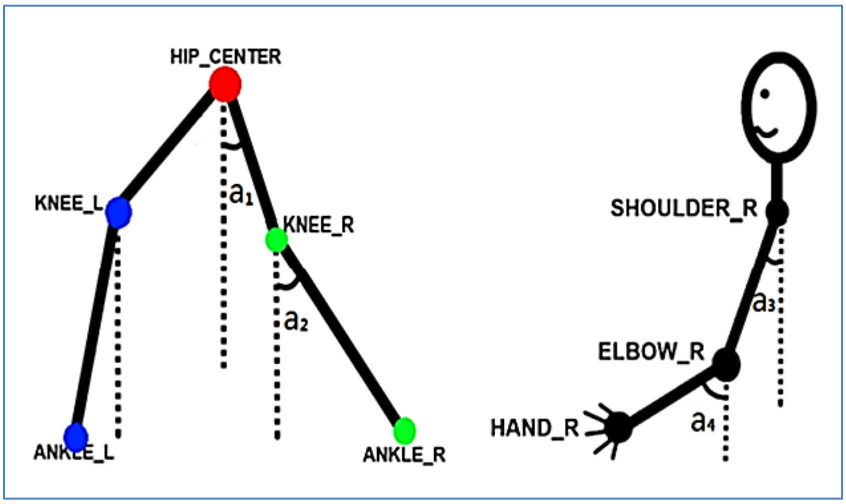


Figure 1. Schematic of dynamic feature

Media Pipe Pose is a ML solution for high-fidelity body pose tracking, inferring 33 3D landmarks and background segmentation mask on the whole body from RGB video frames utilizing our Blaze Pose research that also powers the ML Kit Pose Detection API. Current state-of-the-art approaches rely primarily on powerful desktop environments for inference, whereas our method achieves real-time performance on most modern mobile phones, desktops/laptops, in python and even on the web [7].

In this paper feature extraction depended on distance and angle. The Euclidean distance is the Cartesian technique establishes a straight line distance between locations. The Euclidean distance is defined as the square root of the sum of the squared differences between two points the origin (i) and destination (j) values for all variables, as illustrated in Eq.

$$d_{ij}=\sqrt{\sum_{v=1}^{p}\left(X_{iv}-X_{jv}\right)^{2}} \tag{2}$$

Where: X_{iv} is represents the characteristics of individual i;

X_{jv} is represents the characteristics of individual j;

p is the number of portions in the sample;

v is the number of individuals in the sample [8].

Angle is union of two rays makes an angle with the same starting point; the rays are referred to as the arms. A rotation about P, which defines the magnitude of the angle between two points, can be used to make the rays coincide. In other words, each arm specifies a direction, and the angle size equals the difference between these directions. This difference might be positive in either clockwise or counterclockwise direction Each angle is a distinct dynamic feature [9, 10].

2. Related works

Jiande Sun and et.al in 2018 [11] the second-generation Kinect (2G–Kinect) is utilized to create a 3D skeleton-based gait dataset that combines 2D silhouette pictures from the 2G-Kinect sensor with the corresponding 3D coordinates of the joints. A human walking model is created using this dataset. In the context of the walking model, the length of a few particular skeletons is chosen as the static feature and the swing limb angles as the dynamic feature. Both of these characteristics are confirmed to be view-invariant. Additionally, both static and dynamic feature gait detection capabilities are looked at. In light of the inquiry, a nearest neighbor (NN) method-based view-invariant gait identification strategy is developed. It is based on the matching-level-fusion of the static and dynamic information. On several datasets, comparisons between the proposed technique and the current Kinect-based gait recognition method reveal that the new method performs better in terms of recognition.

Lingxiang Yaoa and et.al in 2019 [12] they proposed Skeleton Gait Energy Image (SGEI), a model-based gait representation, has demonstrated that it is an effective feature addition that improves the resilience and accuracy of gait detection in unrestricted situations with view shifts and clothing changes. SGEI is the averaged spatial model across a whole walking cycle after skeleton extraction and period estimation. Gait characteristics are converted from several viewpoints into a single common subspace using two VTMs that were separately built for SGEI and GEI (Gait Energy Image). By integrating SGEI and GEI with various weights, a discriminative hybrid descriptor is learnt for gait detection. It was determined that this hybrid descriptor performed better than GEI when just using the robustness of the CASIA Gait Database B and effectiveness of feature SGEI have been certified, and as a feature complement.

Ehsan M. Owaidah, and et al in 2019 [13] they had investigated the usage of the Kinect for thobe or abaya-clad Saudis to identify their gaits. The majority of the joints are concealed by these clothes, making gait analysis difficult. The top three joints that might provide the greatest identification results are identified by the algorithm using Kinect, and these joints are subsequently used for gait recognition. Joints' Y coordinates were used as features, and the K-Nearest Neighbor classifier was employed. A video recording of each person was taken using the Kinect sensor. The discovery that the three best joints, when used in combination, provided accuracy above 92 percent on the considered data was an intriguing result of this study. This was done five times for each participant and more than 91% accuracy was attained on that.

Bari and Gavrilova, 2019 [14] the dynamic joint relative cosine dissimilarity and joint relative triangular area feature vectors were used to train the proposed neural network model. Iteratively minimizing the loss of the goal function is achieved by subsequently applying the Adam optimization approach. On two publicly accessible 3D skeleton-based gait datasets captured with the Microsoft Kinect sensor, the performance was assessed. It has been demonstrated through experiments that the suggested neural network architecture, trained with the help of newly added dynamic geometric features, outperforms previous cutting-edge techniques for Kinect skeleton-based gait identification.

Jun and et.al 2020 [15] they With the help of two RNN AEs, a long short-term memory (LSTM)-based AE and a gated recurrent unit (GRU)-based AE, characteristics from skeletal gait data were retrieved (GRU AE). The original skeleton data and other features are contrasted with the RNN AEs' features. By submitting the features to multiple discriminative models (DMs) and comparing the recognition results, they assessed the features. Compared to the original skeleton data and other existing features, the features extracted using RNN AEs are more robust and immediately recognizable. Particularly, the LSTM AE outperforms the GRU AE in terms of performance. Hybrid models, where the RNN AEs' features are supplied to DMs, are superior to single DMs fed with the original skeleton directly demonstrate greater recognition accuracy with fewer learning parameters and training epochs. Therefore, by streamlining arduous procedures and efficiently boosting recognition accuracy, the suggested automated feature extraction technique enhances the performance of skeleton-based abnormal gait recognition.

Kwon and et.al 2021 [16] they performed a comparison of two methods for viewpoint-invariant person re-identification using gait patterns: feature-based and spatiotemporal-based. To identify a person, the first technique uses gait characteristics that are retrieved from time-series 3D joint positions. The second technique classifies a person without extracting gait data by using a neural network called a Siamese Long Short Term Memory (LSTM) network with the 3D spatiotemporal changes of key joint locations in a gait cycle.

They experimented with two open datasets, the MARS and CASIA-A datasets, to validate and contrast these two methodologies. According to the results, the Siamese LSTM beats gait feature-based methods on the MARS dataset by 20% and on the CASIA-A dataset by 55%. The results show that feature-based gait analysis using 2D and 3D pose estimators.

3. Methodology

The general outline of the proposed diagnosis system is shown in figure (2), which shows that the system consists of main stages Gaiting video, 3D key joint position for each frame determine, feature extraction, cleaning features, and identification.

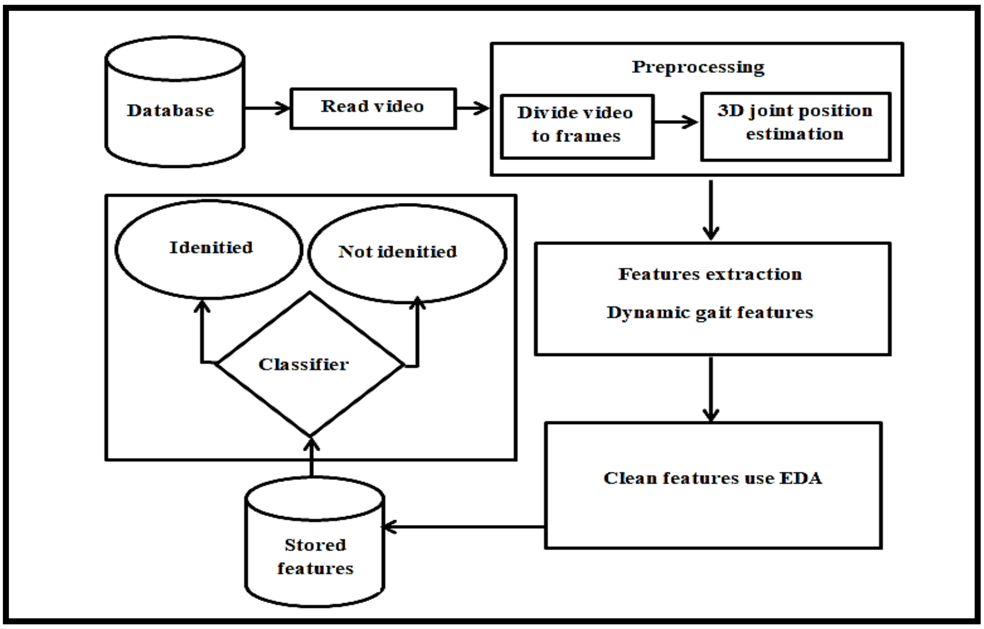


Figure 2. proposed gait system

3.1 Gating video

The video acquisition was performed in real time when person walks in front of the camera. Camera was positioned (1meter) above the ground with the distance to the object of (4 meters), figure (3).

Figure 3. The Location of the Camera to Shoot a Video for Person

We filmed a twenty seven people as samples for the experiment, at two different date and different dressing, each person take six videos. The total number of video used in this research is one hundred and sixty two videos, where each video contains person pedestrian. In the process of frame extraction, all frames in the video were extracted.

3.2 video pre -processing

In this stage video become simple and convert to frame and determined 3D key joint point for each frame by using Media Pipe model. The pose estimation component of current proposal system predicts the location of all 33 person key joint positions figure(4), Start the detector from the first frame of the video to locate the person as well as provide a bounding box around that, then the tracking system takes over and indicates the landmark points inside of that bounding box ROI, then the tracking system continues to operate on any current frame in the video using the previous frame's ROI and only calls the recognition model again when it keeps failing to track the person with high certainty.

3.4 Features Extraction

After 3D pose positions estimation, To identify an individual, walking patterns will be analyzed. Gait is defined as locomotion accomplished by limb and arm movement figure (5)

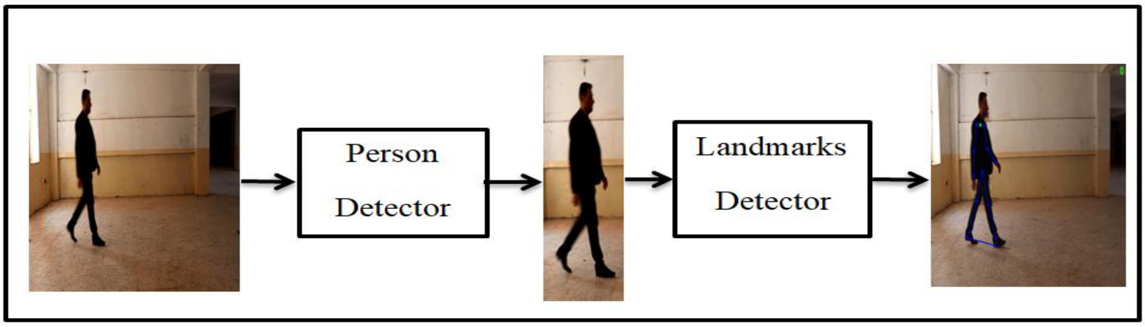


Figure 4. A sample of my study during the walking process

Figure 5. Gait cycle of sample

3.4.1Angles as dynamic features

To get an angle we have two vectors the first from point x to y and x to z use arctan to represented angle. The algorithm (1) explains to extract angles.

Algorithm (1) calculate Angle
Input: Coordinte X, CoordinateY, CoordinateZ
Output: Angle value
Begin **Step1**: Get x1, y1 = Coordinate A.x, Coordinate A.y **Step2**: Get x2, y2 = Coordinate B.x, Coordinate B.y **Step3**:Get x3, y3 = Coordinate C.x, Coordinate C.y **Step4**: Angle = $\arctan\left(\frac{y3 - y2}{x3 - x2}\right) - \arctan\left(\frac{y1 - y2}{x1 - x2}\right)$ **Step5**: if $0 \leq$ angle ≤ 360 **Step6**: angle value **End**

●**Angle of knee and elbow**

Two sets of oscillation angles for the upper limbs were determined, the left elbow angle and the right elbow angle and two sets of angles for the lower limbs, the right knee and the left knee and algorithm (2) explain how to extract angle of hands and legs (left and right). Calculate angle of elbow and knee get the angle of the right knee and the left knee by determine the location of the hip, knee, and ankle and then extract the angle and get the angle of right and left elbow determine the location of shoulder, elbow, wrist then extract the angle.

Algorithm (2) knee and elbow angles
Input: Video
Output: Angles value
Begin **Step 1**: input video **Step 2**: for each frame **Step 3**: get the frame width, height **Step 4**: calculate angle knee get left [lip, knee, ankle]=pose landmark model (each frame) get Right [hip, knee, ankle]=pose landmark model (each frame) feature angle knee left = calculate angle left (hip, knee, ankle) feature angle knee right = calculate angle right (hip, knee, ankle)

Step 5: calculate angle elbow

get left [shoulder, elbow, wrist]=Pose landmark model (each frame)

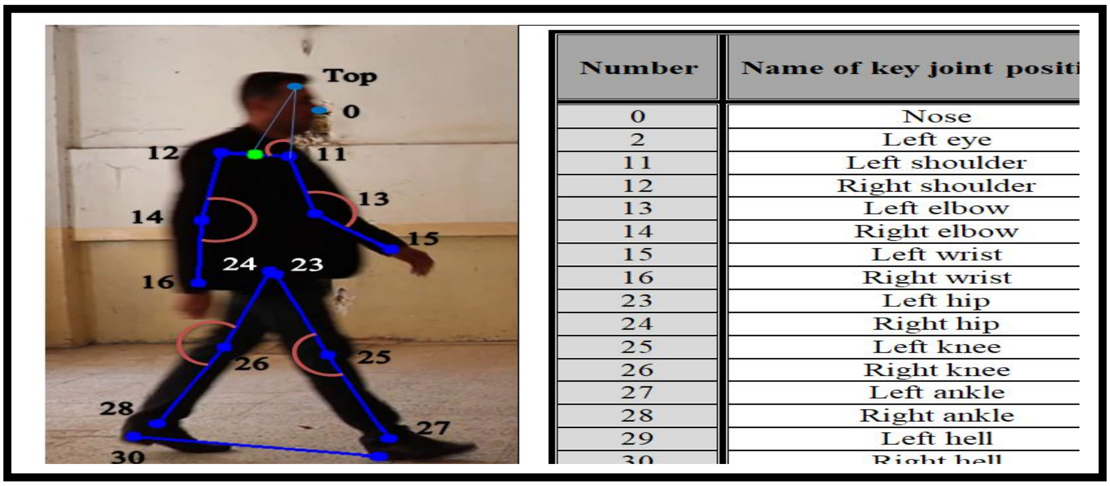
get right [shoulder, elbow, wrist]=Pose landmark model (each frame)

feature left angle elbow = calculate left angle (shoulder, elbow, wrist)

feature right angle elbow = calculate right angle (shoulder, elbow, wrist)

Step 6:saveto csv (features left angle knee, feature right angle knee, feature left angle elbow, feature right angle elbow)

End



Number	Name of key joint positi
0	Nose
2	Left eye
11	Left shoulder
12	Right shoulder
13	Left elbow
14	Right elbow
15	Left wrist
16	Right wrist
23	Left hip
24	Right hip
25	Left knee
26	Right knee
27	Left ankle
28	Right ankle
29	Left hell
30	Right hell

Figure (3) Key joint position

-Calculate Euclidean distance

Euclidean distance is a method that measures the distance between two points. Algorithm (3). The line contains four values, the line consists of (x1, y1) and its end consists of (x2, y2), which are four points entered in the Euclidean equation and the result is a distance value.

Algorithm (3) calculate Euclidean distance
Input: Coordinate A, Coordinate B
Output: Distance between A and B
Begin **Step1**: Get x1, y1 = Coordinate A. x, Coordinate A. y **Step2**:Get x2, y2 = Coordinate B. x, Coordinate B. y **Step3:** distance=$\sqrt{(x1-x2)+(y1-y2)}$ **Step4**: value of distance **End**

-Mid-point

Algorithm (4) calculate mid-point is a method that measures the mid distance between two points. The line contains four values, i.e. the line consists of (x1, y1) and its end consists of (x2, y2), which is four points entered in the mid equation and the result is a specific value.

Algorithm (4) calculate midpoint
Input: Coordinate A, Coordinate B
Output: value of midpoint
Begin **Step1**: Get x1, y1 = Coordinate A. x, Coordinate A. y **Step2**: Get x2, y2 = Coordinate B. x, Coordinate B. y **Step3**: midpoint = $\left(\frac{(x1+x2)}{2}, \frac{(y1+y2)}{2}\right)$ **Step4**:value of midpoint **End**

- **Angles of head drop left**

To find the head drop left angle algorithm (5), first we must find a point in the middle of the forehead and find a point in the middle of the space between the right and left shoulder**,** algorithm (3) explain how to calculate Euclidean distance between nose and left eye. To generation top point use the equation:

Top=[nose, (left eye –distance nose and left eye)],

Algorithm (4) explain how to calculate mid-point between right and left shoulder.

Algorithm (5) Head –drop –left
Input: Video
Output: head drop left
Begin **Step 1**: input video **Step 2**: read each frame **Step 3**: get the frame width, height **Step 4**: calculate distance get [nose, left eye, left ankle]=pose landmark model (frame) distance = calculate Euclidean distance (nose, left eye) **Step 5**: generation new pose landmark top = [nose, left eye – distance] **Step 6**: calculate angle head drop left get [left shoulder, right shoulder]=Pose landmark model (frame) left right shoulder = calculate midpoint (left shoulder, right shoulder) feature angle head drop left = calculate angle(Top, left right shoulder, left shoulder) **Step 7**: save to csv (feature angle head drop left) **End**

●Distance between left heel and right heel

Distance between left heel and right heel is change during gait cycle. Algorithm (6) explain how to calculator distance (length) between left heel and right heel.

Algorithm(6) extraction length between left heel and right heel
Input: Video
Output: distance (length) value
Begin **Step 1**: input video

Step 2: for each frame

Step 3: get the frame width, height

Step 4: calculate distance

get [left heel, right heel]=pose landmark model (frame)

feature distance = calculate Euclidean distance (left heel, right heel)

Step 5: save to csv (Feature distance value)

End

3.4 Cleaning Features

In the cleaning step, use exploratory data analysis to discover a lower and upper range of the feature that reduces the possibility of mistakes while retaining the powerful characteristics that produce high-accuracy diagnostic findings. The features cleansing stage was accomplished in this paper, the Algorithm (7) describes the features cleaning application using (EDA).

3.5 Identification Stage

Identification a person by analyzing the pattern of the gait which is obtained through the previous stages. Every gait image is converted to gait code or pattern to be identification algorithms, a feature has been used as input in the Denes neural network.

3.5.2 Denes neural network

In this paper, the network structure consists of an input layer containing 75, and there hidden layers, the first contains 1000 nodes, the second contains 500 nodes, and thread contains 300 nodes use random weight, finally, the output layer contains 27 nodes representing an output corresponding to 27 classes, using optimizer is SGD.MLP uses a supervised learning technique. During the learning phase, the outputs are compared with the contents of the pre-stored database and the desired results are obtained. Initially the weight is created randomly, weights are iteratively modified. Using the error between the outputs corresponding to the desired output algorithm (8).

Algorithms (7) Remove outlier
Input: data set
Output: new dataset
Begin **Step1**: for each column in data set do **Step2**: get lower range, upper range from column **Step2-1**: Sorted (column) **Step2-2**: Find quantile form column Q1 = column. quantile (0.25) Q3 = column. quantile (0.75) **Step2-2**: Find lower range, upper range IQR = Q3 - Q1 lower range= Q1 - (1.5 * IQR) upper range= Q3 + (1.5 * IQR) **Step3**: update data column **Step3-1**: for each value in column do if value < lower range then new value = lower range(update value) in column if value >upper range then new value = upper range (update value) in column **Step3-2**: end for **Step4**: end for **End**

Algorithms (8) denes neural network Algorithm
Input: Features (75 for each image), Class Label (person or not)
Output: Class the predicted
Begin **Step 1**: Loading x= features **Step 2**: Loading y =class label **Step 3**: [r,c]= size (x). **Step4**: Normalizing by equation x=(x –mean(x)/ standard deviation (x) To scales the data values to be in a specified small range. **Step 5**: Determine the percentage of the training and testing mode by: k= length (y); training input= round (0.8*k); testing=k-training ; **Step6**: model sequential model. add(dense(1000,input_dim=75, activation=’relu’,kernel initializer=’he uniform)) Model.add(dense(500, activation=’ReLu’)) Model.add(dense(300,activation=’ReLu’)) Model.add (dropout(0.2)) Model.add(dense(27,activation=’softmax’)) **Step7**: Use the Classifier for the testing phase to predict testing data to Get the classification results, the person or not **End**

4. Supervised Learning Algorithm dense NN

In this stage the dense neural network is calculated on the feature is used to test the feature which is dance neural network, dense neural network is calculated on the feature that has been extracted by earlier stage (6 in dynamic gait feature) each frame in and this will represent the video of human gait action then the classification is based on finding dense neural network.

4.1 Results for dynamic gait features with dense NN

We have six features were samples per category. In figure (4) can see that the total of each row in this matrix is also six. The value of column number five in row number five of this matrix is 4, the column number twenty two have value 1and the column number twenty five have value 1. That is, all four of the class five test samples were (properly) identified as such (column 5), and 1 sample is (incorrectly) identified as belonging to class 22 (column 22 has value=1) and 1 sample is (incorrectly) identified as belonging to class 25 (column 25 has value=1). And yet another. This is how we read a Confusion matrix. As a result, it is acknowledged that categorization is excellent if diagonal values are greater than no diagonal values. Figure (5) show the evaluating of classification performances, the value of precision and recall and f-score is equal in each class that mean the algorithm classified equal amounts of the data set as false positives and as false negatives.

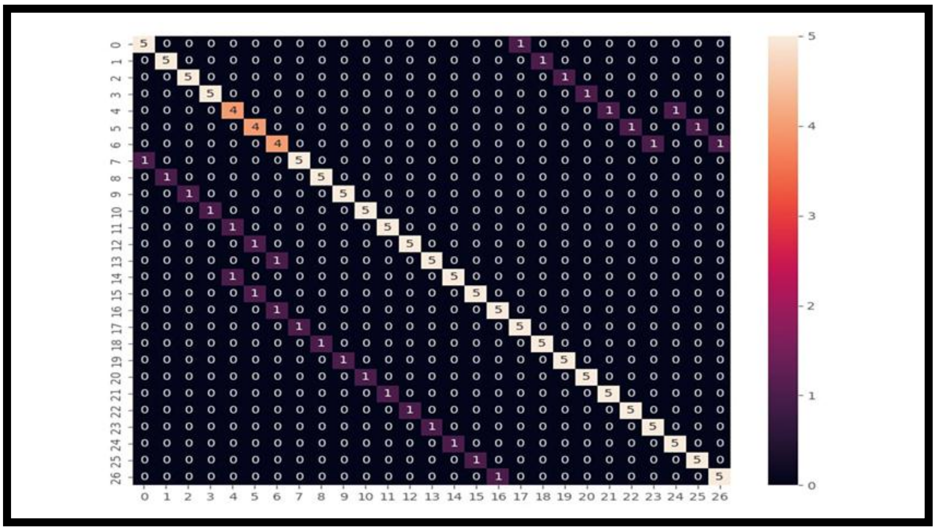


Figure (4). Confusion matrix for dynamic gait feature with accuracy 81%

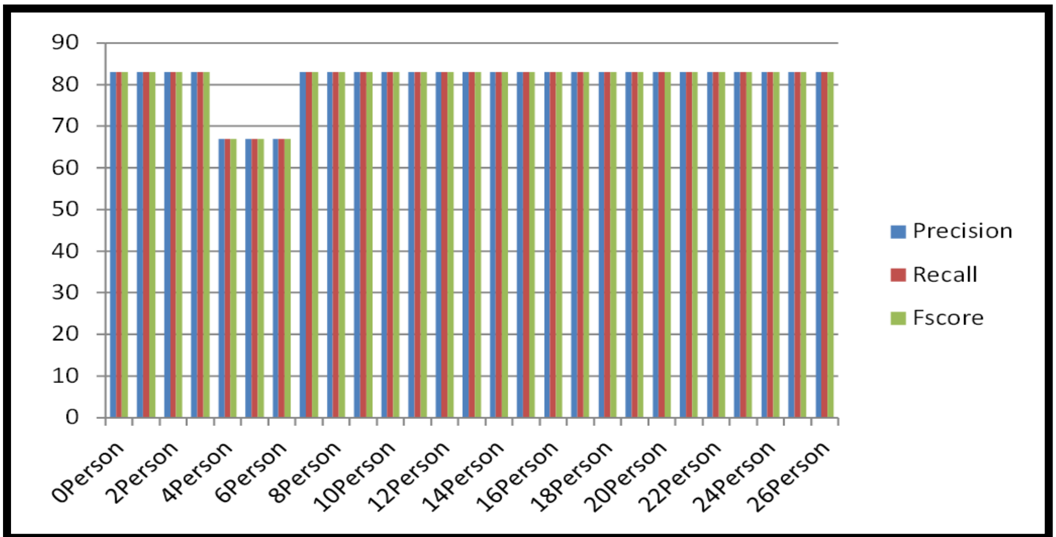


Figure 5. Results of the precision, recall and F-score

5. Conclusions

Identification human by using dynamic gait features and dense neural network. The proposed method show the exploratory data analysis is effective to remove outlier data. The results showed that the dense identification achieves good accuracy in the identification with dynamics gait feature 81%.

6. Reference

[1] Bobick, A. F., & Johnson, A. Y. (2001). Gait recognition using static, activity-specific parameters. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* (Vol. 1, pp. I-I). IEEE.

[2] Wan, C., Wang, L., & Phoha, V. V. (Eds.). (2018). A survey on gait recognition. *ACM Computing Surveys (CSUR)*, *51*(5), 1-35.

[3] Han, J., & Bhanu, B. (2005). Individual recognition using gait energy image. *IEEE transactions on pattern analysis and machine intelligence*, *28*(2), 316-322.

[4] Ahmed, M., Al-Jawad, N., & Sabir, A. T. (2014). Gait recognition based on Kinect sensor. In Real-Time Image and Video Processing 2014 (Vol. 9139, pp. 63-72). SPIE.

[5] Ziad M. Abood, Karam, G. S., Abbas, F. I., Kadhim, K. K., Karam, N. S, (2018), An enhanced approach for biomedical image restoration using image fusion techniques, AIP Conference Proceedings, 1968.

[6] Jiang, S., Wang, Y., Zhang, Y., & Sun, J. (2014, November). Real time gait recognition system based on Kinect skeleton feature. In *Asian Conference on Computer Vision* (pp. 46-57). Springer, Cham.

[7] Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., & Grundmann, M. (2020). Blazepose: On-device real-time body poses tracking. *arXiv preprint arXiv: 2006.10204*.

[8] Ziad M. Abood, Karam, G. S., Haleot, R. E. (2017). Face Recognition Using Fusion of Multispectral Imaging, 2017 2nd Al-Sadiq International Conference on Multidisciplinary in IT and Communication Science and Applications, AIC-MITCSA 2017, pp. 107-112

[9] Pfister, A., West, A. M., Bronner, S., & Noah, J. A. (2014). Comparative abilities of Microsoft Kinect and Vicon 3D motion capture for gait analysis. *Journal of medical engineering & technology*, *38*(5), 274-280.

[10] Clements, D. H., & Battista, M. T. (1990). The effects of Logo on children's conceptualizations of angle and polygons. *Journal for research in mathematics education*, *21*(5), 356-371.

[11] Sun, J., Wang, Y., Li, J., Wan, W., Cheng, D., & Zhang, H. (2018). View-invariant gait recognition based on kinect skeleton feature. *Multimedia Tools and Applications*, *77*(19), 24909-24935.

[12] Yao, L., Kusakunniran, W., Wu, Q., Zhang, J., Tang, Z., & Yang, W. (2021). Robust gait recognition using hybrid descriptors based on skeleton gait energy image. *Pattern Recognition Letters*, *150*, 289-296.

[13] Owaidah, E. M., Aloufi, K. S., & Alkhatib, J. H. (2019, May). Gait Recognition for Saudi Costume Using Kinect Skeletal Tracking. In 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS) (pp. 1-5). IEEE.

[14] Bari, A. H., & Gavrilova, M. L. (2019). Artificial neural network based gait recognition using kinect sensor. IEEE Access, 7, 162708-162722.

[15] Jun, K., Lee, D. W., Lee, K., Lee, S., & Kim, M. S. (2020). Feature extraction using an RNN autoencoder for skeleton-based abnormal gait recognition. IEEE Access, 8, 19196-19207.