sciendo

VOLUME 17, N° 3 2023
Journal of Automation, Mobile Robotics and Intelligent Systems

Improved Competitive Neural Network for Classification of Human Postures Based on Data from RGB-D Sensors

Submitted: 5th January 2023; accepted: 18th April 2023

Vibekananda Dutta, Jakub Cydejko, Teresa Zielińska

DOI: 10.14313/JAMRIS/3-2023/19

Abstract:
The cognitive goal of this paper is to assess whether marker-less motion capture systems provide sufficient data to recognize human postures in the side view. The research goal is to develop a new posture classification method that allows for analysing human activities using data recorded by RGB-D sensors. The method is insensitive to recorded activity duration and gives satisfactory results for the sagittal plane. An improved competitive Neural Network (cNN) was used. The method of preprocessing the data is first discussed. Then, a method for classifying human postures is presented. Finally, classification quality using various distance metrics is assessed. The data sets covering the selection of human activities have been created. Postures typical for these activities have been identified using the classifying neural network. The classification quality obtained using the proposed cNN network and two other popular neural networks were compared. The results confirmed the advantage of cNN network. The developed method makes it possible to recognize human postures by observing movement in the sagittal plane.

Keywords: *Human motion, Posture classification, Human activity, Competitive neural network, Classifier*

1. Introduction

An activity (e.g., cooking, exploring, searching) includes a sequence of actions. Kinematic features extracted using human body models are used to identify human actions. Knowing that actions are their elementary components makes it possible to anticipate human activities. At the lowest abstraction level, human actions are recognised by identifying the sequence of the motion primitives that compose them, which can be addressed as the recognition of the posture types. The motion primitive is the small 'uniform' fragment of motion associated with the specific posture. In humanoid robots (and in the case of a human), the movement is carried out at a certain posture, and this does not mean that the robot (or human) is 'frozen' in place.

The posture is associated with the features of kinematic configuration (e.g., elbow joint oriented 'in', elbow joint oriented 'out', sharp or open-angle is kept in the knee joint, etc.). It does not mean that the angular positions are constant; they undergo the changes to such an extent that the specific posture is kept. Many different criteria are used to recognise different postures. The robots move to withhold some posture (please see the robotics textbooks and classic publications, e.g., [24, 25]). With such interpretation, the activity, motion primitive, and posture are not very distinct. If the activity is realised with one posture or by more postures, it depends on the specificity of the activity and on the criterion or resolution used to distinguish the postures.

Research on postures recognition is needed to synthesize the movement of humanoid robots. Operating in an unstructured and unfamiliar environment, these robots are expected to plan their actions using the knowledge of the sequence of postures observed in humans [29]. Posture recognition is also useful for recognizing and predicting human activities. Some researchers use the identification of posture sequences for this purpose [14]. In our other paper, we proposed a semantic database for inferring current activity from a sequence of actions [8, 9]. Presented research may also be helpful in the diagnostic analysis of human movement [22], including in detecting and studying movement disorders.

1.1. Research Challenges and Contribution

One of the essential problems in the field of robotics supporting human activities is an automated inference about what and how a human is doing. Moreover, assisting robots should not stress a person with their appearance, unusual equipment [29], or strange posture. Hence, there is a need for registration and analysis of human movement [10]. The latest technological advances, including higher-resolution depth sensors and more efficient methods of motion data processing, make it easier to track the movement of parts of the human body.

One of the goals of this work is to contribute to the research on the motion planning of a humanoid robot using the concept of learning by observation. The robot is not expected to imitate the motion of a human, but it will learn a sequence of key postures taken by a human performing an activity. The robot will perform these postures in its own way as the kinematics, dynamics, and features of the actuating system are different from a human. The activity will be created by transformations between the postures that make the sequence. We can assume greater 'resolution' of such sequences with many postures, but then converting them to robot postures requires considerable effort.

On the other hand, using a small number of postures will leave a lot of 'freedom' to the humanoid robot motion planning system, which can lead to strange (postures not typical for a human being) intermediate postures. Therefore, it is logical to identify the key postures with the appropriate resolution. It is suitable to choose those that are not too similar to each other. In this work, posture is associated with an action. This means that ongoing human activity can be anticipated through sequences of identified postures. Therefore, developing a method that can be used for such a purpose is the second goal of this paper.

This article proposes a novel approach to inferring human postures using a simplified model of the human body and taking into account the data from the RGB-D sensors. Human movement is mainly observed in the sagittal plane because, in this plane, the most visible changes in posture are made during basic activities such as lifting, carrying, reaching, and collecting. Finally, machine learning methods are used, considering the positions of relevant human body points to classify (to recognise) human postures [28].

The research contribution of this work is as follows:

1) a body posture classification procedure using data delivered from RGB-D sensors was proposed,
2) following [13] this work introduces a modified method to approximate the missing data and reduce measurement errors. The normalization of body coordinates in the data pre-processing stage was also done,
3) the most appropriate distance metrics were indicated for the shallow competitive neural network (cNN) classifying human postures,
4) the developed classification method (leading to posture recognition) was assessed, and compared with the results obtained using other classifying networks, and the advantages of this method were demonstrated,
5) the datasets with observations of various human activities in the sagittal view were created and shared.

The remaining part of this paper presents the contribution in detail. In Section 2, we discuss the previous works in the area of recognising human actions/activities. Section 3 describes the data processing method; next, the proposed method is introduced and tested, considering different distance metrics. Section 4 presents the evaluation of classification quality. Finally, section 6 gives the conclusions and indicates the directions of future work.

2. Related Work

The state-of-the-art works on human postures identification (which also means human action/activity recognition) focus on relations between body points position that define the body posture in each moment. Rogez et al. [19] proposed an end-to-end Localization-Classification-Regression Network (LCR Net) for 2D and 3D pose estimation using video images. The probabilistic method to predict human movement trajectories was successfully investigated by Dutta et al. [9] considering the needs of robot helpmates.

Various works [2, 16] have sought to generate 3D pose estimation from 2D keypoints. Hou et al. [12] presented a multi-channel network based on graph convolutional networks for skeleton-based action recognition. In the work [5], the method for postures recognition in the frontal plane using rather time-consuming coordinates transformation and features extraction was proposed. Recent advances in the imaging techniques offered by high-quality depth sensors that perform well despite lighting and color variations [17, 20] enable the easy capture of human posture.

Lately, the shift from conventional statistical methods to methods based on machine learning has also been noticed. So, using machine learning tools makes it possible to capture complex relationships within recorded datasets [16]. Building the benchmark RGB-D data sets [11, 23], human body joints are often registered in the frontal view using the multi-camera setup and arranged environments. One of the possible approaches for postures recognition is the method of applying the classifying neural networks. Apart from differences in scale and level of abstraction, one of the main difficulties in using neural networks for recognising postures in a 3D space is the relatively limited number of training samples, especially concerning the data from the observations taken in the sagittal plane. These shortcomings and the need for human assisting robots were the motivations for undertaking the presented work.

3. Materials and Methods

3.1. Recording Setup and Constitution of the New Data-set

The movement of actors performing complex activities was recorded in the sagittal plane and at an angle of 50° as shown in Figure 1. The created data set (WUT-22) covers the recording environments' diversity, taking into account the cultural background. Our dataset was recorded in two different environments – the first 10 activities were performed at Warsaw University of Technology, Poland, and the next 10 activities were recorded at Waseda University, Japan, respectively. For data recording, two modern ZED RGB-D cameras with ZED Body Tracking SDK software or two Azure RGB-D cameras with Azure Microsoft Body Tracking SDK software were used [17].

The relationship between the data from the RGB-D image and the coordinates of the human body in three-dimensional space was established [27]. Twelve healthy people aged 24 − 35 years, with a height of 151−187 cm and weight of 65−85 kG, took part in the study with their consent. Twenty activities were performed in a random order. For each involved person performing a specific role, the activity was recorded 3 times, which, with a total of 4 people, gave a total

Table 1. List of recorded activities and their parameters

Activity	Description	No. of objects	No. of postures*	No. of persons
1	Pick up a heavy object from the floor and place it on the table	2	4	1
2	Pick up a light object from the floor and place it on the table	2	4	1
3	Give the other person a bottle of water	3	6	2
4	Give the other person a cup of coffee	2	6	2
5	Take a heavy box from kitchen shelf and place it on the table	1	4	1
6	Clean the wall	4	11	1
7	Pick up the phone and answer the call	3	5	1
8	Take the book from trolley and place it on the table	5	4	1
9	Drag the book cart to another location	3	3	1
10	Push cart with objects near the table and organise them	3	3	1
11	Pick up boxes from the wardrobe and place them in order	3	12	1
12	Greet each other in traditional Japanese style	1	9	2
13	Making a cup of tea and sit on the chair	4	10	1
14	Bring the objects in a cart and arrange them in order	4	11	1
15	Imitate food preparation	3	5	1
16	Offer a cup of coffee to the guest	4	12	2
17	Greet the guest and prepare a cup of coffee for the guest	2	21	2
18	Organize objects with the help of a colleague	5	25	2
19	Have a meal together with a colleague	3	41	2
20	Offer a cup of coffee to the guest and drink together	4	21	2

* indicated by cNN

of 12 records. The nine randomly selected series of recordings of each activity performed by each person participating in the experiment were selected for training and the remaining three for testing. Data for two different viewing angles were recorded simultaneously. Different camera arrangements were used (see Fig. 2(b)) to test the resistance of the trained neural network to scene variability. Activities included both – short and long sequences of actions.

Both marker-less motion recording sensors were set to a resolution of 1920 × 1080 pixels, with a recording speed of 15 frames per second and a 120° field of view. The minimum depth range was 20 cm. Besides the RGB-D camera, each system includes an accelerometer, gyroscope, barometer, magnetometer, and temperature sensor; therefore, it can be used for many purposes. Appropriate software has been developed, enabling a simplified representation of the human body. Out of the 34 (ZED) or 32 (Microsoft Azure) human body points seen by the camera, 19 points were selected as essential for posture expression (see Fig. 2(a)). Following [26], the sensors were synchronized to collect motion data simultaneously. Finally, the human motion database (WUT-22) was created and used to elaborate the classification method.

Table 1 lists significant parameters involved during the data set creation, together with the number of postures assigned after the neural network training. Depending on the scenario, 1 to 5 objects were used (see third column). The last column lists the number of persons participating in the activity. The fourth column gives the overall number of postures involved in the activity, including those that are repeated several

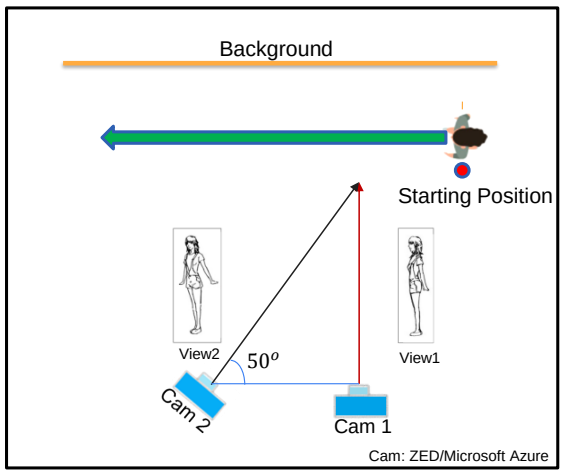


Figure 1. Camera setup in the laboratory environment (top view)

times. The given number of postures is according to the decision of the neural network. In the final version, the cNN had 20 outputs, meaning 20 postures were expected. The column lists how often the postures appeared in the sequence, which means the repeated postures are counted. Therefore, if the recording is longer and consists of repetitive actions, the number of listed postures is greater.

3.2. Data Recording and Pre-processing Stage

The initial data processing stage was carried out using inspirations from the literature, e.g., [13]. Developed software provides the ability to obtain 2D or 3D data describing the position of the human body points. Software associated with used RGB-D cameras has skeleton tracking systems that, based on RGB-D data, automatically provide coordinates of the so-called joints (points of the human body) following a generally accepted standard. These coordinates are the raw data later processed, i.e. filtered and filled with missing points, as described below.

3D information used the (x, y, z, c) data. The x, y, z coordinates were transformed from the world reference frame into the local reference frame attached to the waist. The axes of this frame keep identical and constant orientation as the world reference frame. All positions were expressed in millimeters. Additionally, c is the confidence score (natural number) ranging $\langle 0, 100\rangle$ for ZED and $\langle 0, 3\rangle$ for Kinect Azure. If c falls below the given thresholds, which means no detection' or 'occlusion', the joint information is considered inaccurate and, therefore, approximated considering its neighbouring data. The following parameters were defined:

K – number of frames in the video (it is equal to the number of rows in the recorded CSV file containing the positions data),

I – number of measurement points (they are commonly addressed as the joints) together with their data. For ZED, it is 34 points, where each point data are: x, y, z, c. For Azure, it is 32 points with the x, y, z, c data in each frame, which makes one row the data file, and as it was mentioned c is the confidence score and $c \in \{1, 2, \dots, 100\}$ for the ZED sensor, $c \in \{0, 1, 2, 3\}$ for the Azure sensor,

c_{thr} – the threshold, for ZED $c_{thr} = 80$, and for Azure $c_{thr} = 2$,

bf – before, af – after, the indexes indicating, for each i-th joint accordingly, the closest frame with a valid reading. These frames are the nearest neighbours before (k_{bf}) and after (k_{af}), the k-th frame for which the reading is invalid.

Next, the joint filling process is applied. For clarity of notation lets us denote by ${}^{k}x_i, {}^{k}y_i, {}^{k}z_i, {}^{k}c_i$ the data for i-th joint from k-th frame. The procedure is described as **Algorithm 1**.

Algorithm 1 Filling the missing data

First frame:
$k = 1, i = \{1, \dots, I\}$.
For each i-th joint for which ${}^{k}c_i \le c_{thr}$ find k_{af} for which ${}^{k_{af}}c_i \ge c_{thr}$
and substitute
$({}^{1}x_i, {}^{1}y_i, {}^{1}z_i, {}^{1}c_i) = ({}^{k_{af}}x_i, {}^{k_{af}}y_i, {}^{k_{af}}z_i, {}^{k_{af}}c_i)$
Internal frames:
do
$k = k + 1, i = \{1, \dots, I\}$
For each i-th joint for which ${}^{k}c_i \le c_{thr}$ find:
1) k_{bf} for which ${}^{k_{bf}}c_i \ge c_{thr}$ and
2) $k_{af} \le K$ for which ${}^{k_{af}}c_i \ge c_{thr}$.
if $k_{af} \neq Null$ **then** substitute
${}^{k}a_i = \frac{{}^{k_{af}}a_i - {}^{k_{bf}}a_i}{k_{af} - k_{bf}}(k - k_{bf})$,
$a = x, y, z$ accordingly.
end if
if $k_{af} = Null$ (there is no frame in the remaining part of the recording containing valid reading)
then substitute
$({}^{k}x_i, {}^{k}y_i, {}^{k}z_i, {}^{k}c_i) = ({}^{k_{bf}}x_i, {}^{k_{bf}}y_i, {}^{k_{bf}}z_i, {}^{k_{bf}}c_i)$
end if
while $(k \le K)$
Last frame:
$k = K, i = \{1, \dots, I\}$.
For each i-th joint for which ${}^{k}c_i \le c_{thr}$ find k_{bf} for which ${}^{k_{bf}}c_i \ge c_{thr}$.
and substitute
$({}^{k}x_i, {}^{k}y_i, {}^{k}z_i, {}^{k}c_i) = ({}^{k_{bf}}x_i, {}^{k_{bf}}y_i, {}^{k_{bf}}z_i, {}^{k_{bf}}c_i)$

After filling the missing data, the motion trajectories of the body points were filtered using the Savitzky-Golay [6] filter, which smoothes the data without distorting it. Because biological motion trajectories are naturally smooth, the use of such filtering reduces the noise, causing the abrupt nature of changes in recorded data.

Before feeding them to the artificial neural network, the coordinates were normalized using the following relation:

$$ {}^{k}a_i^{nrm} = \frac{{}^{k}a_i - {}^{k}a_i^{min}}{a_i^{max} - a_i^{min}} \tag{1} $$

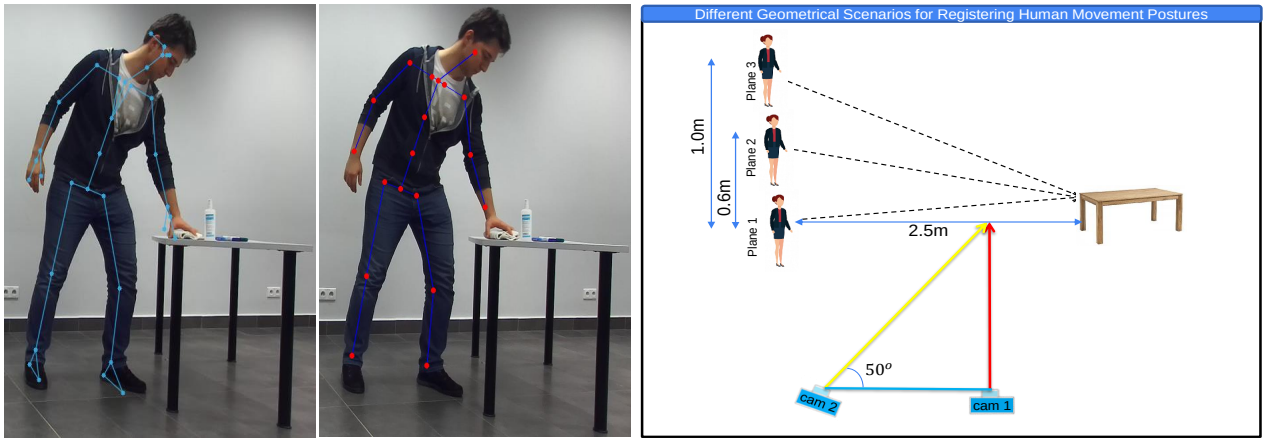


Figure 2. Illustration of data points and applied cNN: (a) view of available points (left) and considered points (right), (b) geometrical scenarios for registering human movement postures

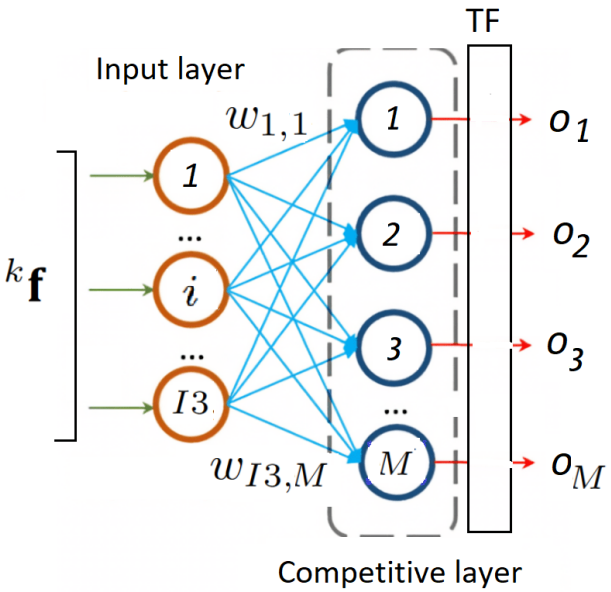


Figure 3. Illustration of the proposed cNN architecture

Where a_i^{min} and a_i^{max} $(a = x, y, z)$ denote the minimum and maximum value of the appropriate coordinate of i-th data point in the whole training set, $^k a_i^{nrm} \in \langle 0, 1 \rangle$ is the normalized value.

3.3. Neural Network

This section describes the implemented neural network by the authors, incorporating an adaptation framework. Adaptation means the selection of the most suitable distance measure. The measure concerns the distances between the weight and input data vectors. Different distance metrics were considered. According to the literature [1], iterative selection of the most appropriate distance metrics is the clue to adaptive learning. In this process, the quality of learning is tested depending on different distance measures. Such measures are finally applied in the cNN that corresponds to the best outcome.

Used cNN is a single-layer neural network in which each output neuron is connected with all inputs. For the purpose of cNN description, let us assume that each input vector containing $I3 = I \cdot 3$ elements (which are normalized coordinates) is denoted by $^k\mathbf{f} = [\, ^k f_1, \, ^k f_2, .., \, ^k f_i, .., \, ^k f_{I3} \,]$, where $k = 1, ..., K$ is the frame index. The number of output neurons equals M – the maximum number of postures plus one. The additional output is left for unrecognised postures, which means the postures which were not identified (and counted) by the human expert. The output neurons compete with each other to become activated after the input data vector is applied [1, 4]. The competition concerns the distance measure.

Lets us denote by $\mathbf{w}_m$ the vector of weights $[\, w_{1,m}, \, w_{2,m}, ..., \, w_{I3,m} \,]$ associated with the $m-th$ output neuron. For each output neuron, the distance measure $d_m(^k\mathbf{f}, \mathbf{w}_m)$ is calculated according to the formulas described below. Let's denote by k_w the winning neuron, and it is such a neuron for which the distance measure $d_{k_w}(..)$ between the input vector and weights vector is the smallest:

$$d_{k_w} = \min_{m=1,..,M} d_m(^k\mathbf{f}, \mathbf{w}_m) \tag{2}$$

During the learning phase, the weights vector of the winner is updated according to the learning rule:

$$\mathbf{w}_{k_w}(n+1) = \mathbf{w}_{k_w}(n) + \alpha(^k\mathbf{f} - \mathbf{w}_{k_w}(n)) \tag{3}$$

n is the learning iteration (epoch) number, α is the learning rate. The rule is called '*winner takes all*'.

Knowing that the performance of the cNN depends, among others, on the initialization method of weights, the different initialization options were tested (the weights equal to the random values, the weights equal to the mean values of the training data, and the weights initialized using the Gaussian distribution). It was observed that the weights initialized using Gaussian distribution offer the best classification performance:

$$w_{i,m} = \frac{1}{\alpha\sqrt{2\pi}} \exp\left(\frac{-(rd - \mu_i)^2}{2\sigma_i^2}\right) \tag{4}$$

where rd is a random number from the range $\langle 0, 1\rangle$, μ_i $(i = 1, .., I3)$ is the mean value calculated taking into account all normalized training data applied as the $i-th$ input $(\mu_i = (\sum_{k=1}^{K} {}^{k}f_i)/K)$, σ_i is the standard deviation of these data. After training, each output neuron represents a cluster in the input dataset, meaning that its weight vector makes the center of that cluster [21].

Figure 3 shows the general architecture of a cNN. An input vector ${}^{k}\mathbf{f}$ of normalized data is supplied to each m-th output neuron. Each neuron has its weights vector $\mathbf{w}_m$ selected in the training process, as described. These weights are 'frozen' in the testing phase. The distance d_m between the current input vector and the weights vector is calculated. This distance is determined using the adopted metric. The distances determined for all neurons are compared, and the neuron with the smallest distance wins. The transfer block (TF) generates outputs of 0 for all neurons except the winner, whose output is 1. We will not give the formulas for all distance metrics tested in this work as they are easily available. To focus the attention, we give the formula for angular (Cosine) distance, which was selected as the most appropriate, as described later. *Cosine distance* has often been used to classify high dimensional data. It is the dot product of the vectors divided by the product of their lengths. The *Cosine* metric does not depend on the magnitudes of the vectors but only on the angle between them.

The *Cosine distance* between ${}^{k}\mathbf{f}$ and $\mathbf{w}_m$, is defined as:

$$d_{cos} = 1 - \frac{{}^{k}\mathbf{f} \cdot \mathbf{w}_m}{\|{}^{k}\mathbf{f}\| \, \|\mathbf{w}_m\|} \tag{5}$$

where $\mathbf{w}_m$ denotes the weights vector of the m-th neuron of the competitive layer.

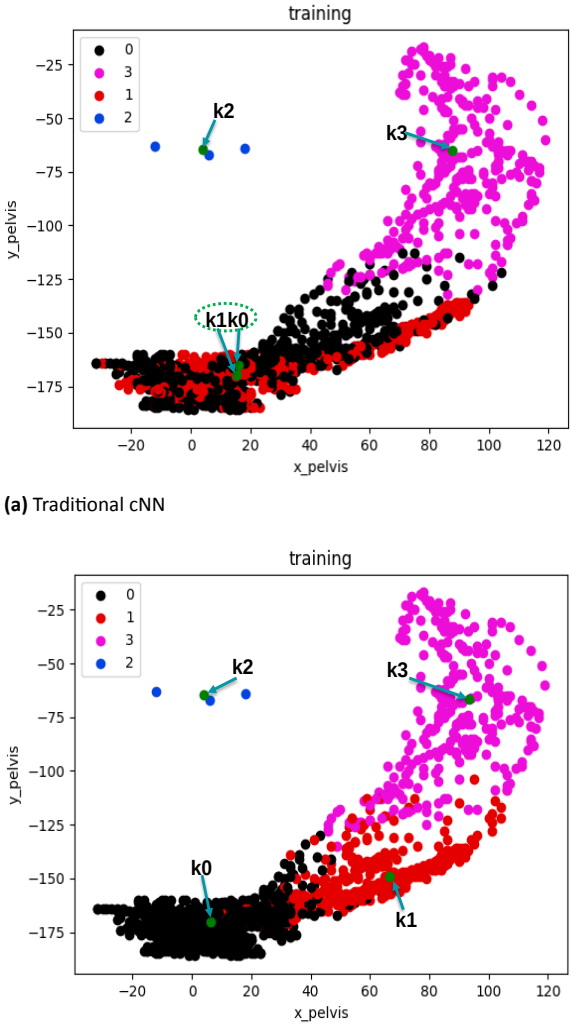
Avoiding overlapping clusters (which means the bad grouping of inputs [18]), a rule of repulsion was applied, 'pushing' the winner away from the rest of the neurons. Thus, during the training, after updating the weights of the winning neuron (neuron k_w), such weights $w_{i,m}$ are moved away from the winner k_w for which the absolute difference between $w_{i,m}$ and the weight w_{i,k_w} is less than the defined Δ threshold:

$$|w_{i,m} - w_{i,k_w}| < \Delta \tag{6}$$

Such modification prevents the clusters nearer than the threshold Δ, and the modification rule applies the repulsion rate β, also called the conscience bias learning rate. Denoting by $w_{i,r}$, the neurons which are within Δ range from the winner w_{i,k_w}, the following formula is applied:

$$w_{i,r}(n+1) = w_{i,r}(n) - \beta(w_{i,k_w}(n) - w_{i,r}(n)) \tag{7}$$

In our research, the performance of the cNN with repulsion was compared with the conventional cNN. Both networks were trained using the same distance metric (in this case, Euclidean distance) for the same input data set. As seen in Figure 4, the repulsion strategy offered better placement of the central points of the clusters. These points (marked as green dots) are well separated (Fig. 4(b)).



(a) Traditional cNN

(b) Proposed method (cNN + repulsion rule)

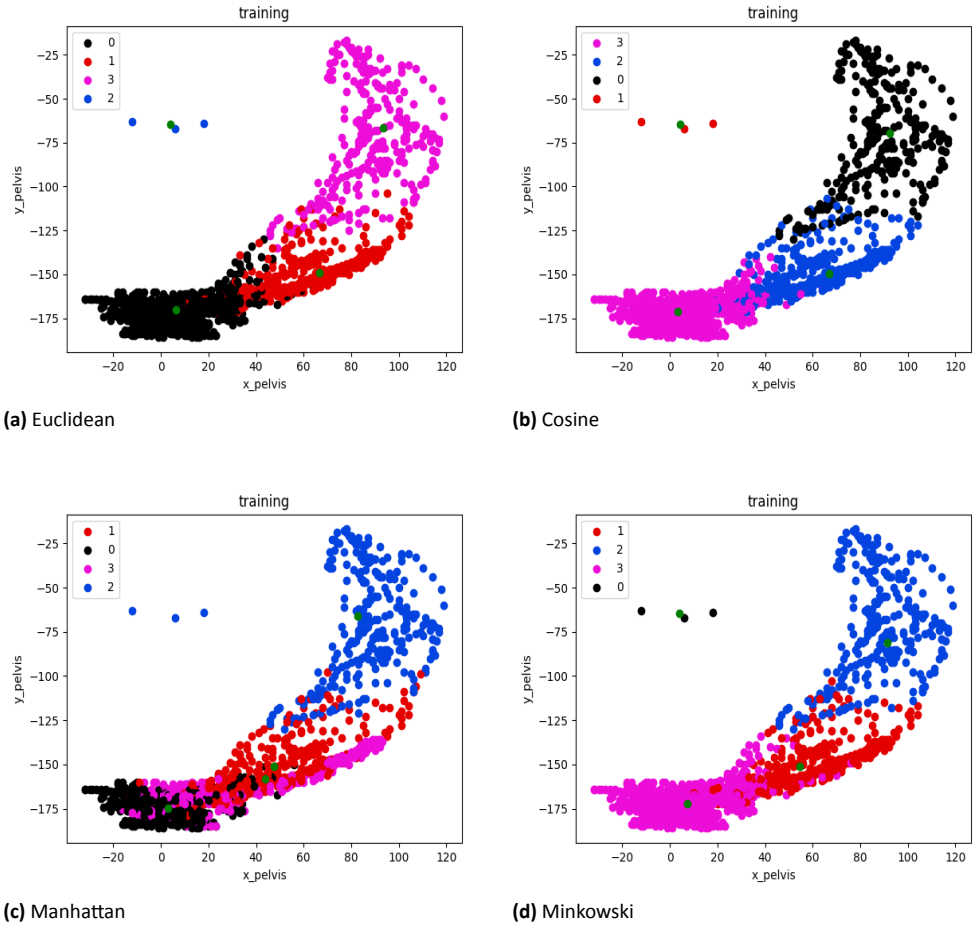
Figure 4. Distribution of the winning neurons for the traditional cNN (upper part) and cNN with repulsion (lower part), $k0$, $k1$, $k2$, $k3$ are denoting the centers of each class for Activity 1

3.4. Implementation Details

The collected data concerned 32 or 34 points of the human body, but the data from 19 points were selected as a set sufficient to describe the body posture. Given that each point has 3 coordinates, it brings $I3 = 57$ inputs to the neural network, normalized with (1). Thus, each camera provides the input vector ${}^{\mathbf{k}}\mathbf{f} = [{}^{k}f_1, {}^{k}f_2, \ldots, {}^{k}f_{57}]$ for each kth sample. During the training, input vectors from both cameras were supplied to the neural network, but only data from the camera placed in the sagittal plane were used for the tests. This arrangement made it possible to observe the resilience of the neural network to limited data.

According to the literature [18], the value of β, the repulsion rate, should be less than α, the learning rate.

As it was already mentioned, the number of outputs of the neural network was by one greater than the number of expected postures. The postures 'labelling' were done after the neural network grouped the data. This was performed in the learning phase, linking the time interval of input data recording and the



(a) Euclidean **(b)** Cosine

(c) Manhattan **(d)** Minkowski

Figure 5. The scatter plots showing the distribution of the winning neurons for different distance metrics for Activity 1

corresponding active output of NN with the skeleton diagrams visible in the video. Thus, by observing what posture is maintained in this time range, a label (in other words, semantic meaning) was assigned to this posture. In this way, it was detected which output indicated which posture, which allowed the recognition of postures during the testing phase. Whether an action is carried out with one or more postures depends on the specificity of the action and the resolution used to distinguish the positions. Before training, a human expert suggested the number of expected postures. In the case of all 20 activities, this number was 19, with the added 1 bearing in mind an unidentified case. Therefore, the number of outputs of the neural network was set to 20.

3.5. Repulsion Rule and Selection of the Distance Metric

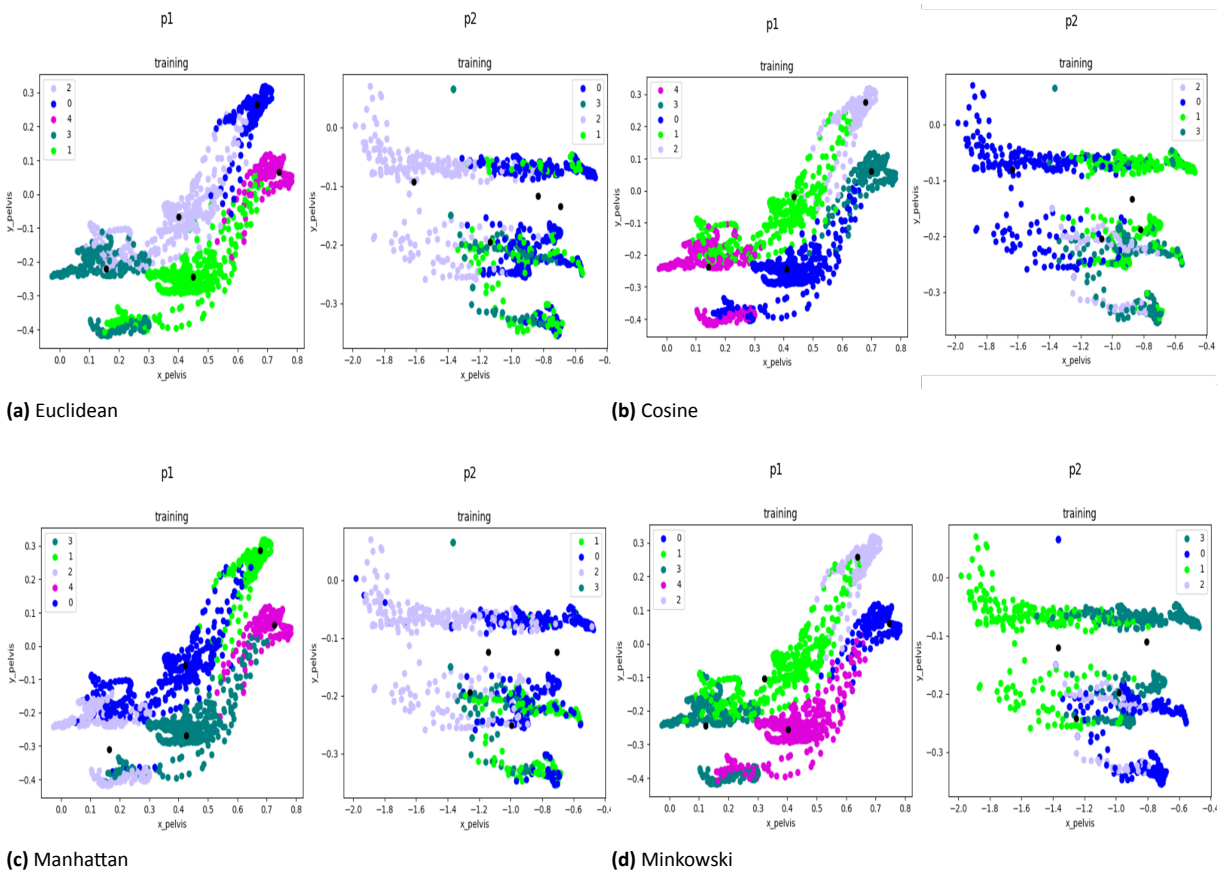
To illustrate the role of repulsion, only one activity was considered, namely Activity 1 (pick up a heavy object from the floor and place it on the table). In this case, a cNN with four outputs was implemented (for three different postures expected by the human expert). Additional output was defined to indicate one more or an incorrectly recognized posture. The cNN with *Euclidean* distance was trained.

The value 0.007 was selected as the learning rate α, and the repulsion rate β was 0.0001. The number of iterations during the training phase was limited to 100 after observing that the minimum distance error remained essentially constant after $70-80$ iterations.

Figure 4 illustrates the advantage of using the rule of repulsion. The upper part of the figure shows the distribution of winning neurons obtained in the training phase for the neurons updated only with the formula (3), and the lower part shows the result obtained using the rules of repulsion (6,7). Each dot represents the winning neutron position obtained for each data frame. As can be seen, repulsion prevents class overlapping, and clusters are clearly visible. We focus on the XY (sagittal) plane data in this and other figures. This is the plane that best shows changes in posture. However, the full set of coordinates (that means x, y, z) was used for training and testing.

After investigating the repulsion role, the most appropriate distance metric was chosen. The widely used *Euclidean, Manhattan, Cosine, Minkowski, Hamming*, and *Mahanalobis* distances were applied in the first stage. Based on the results (not described in this article), four best-performing metrics were selected for the next stage of investigations, namely: (a) *Euclidean* distance, (b) *Cosine* distance, (c) *Manhattan* distance, and (d) *Minkowski* distance.

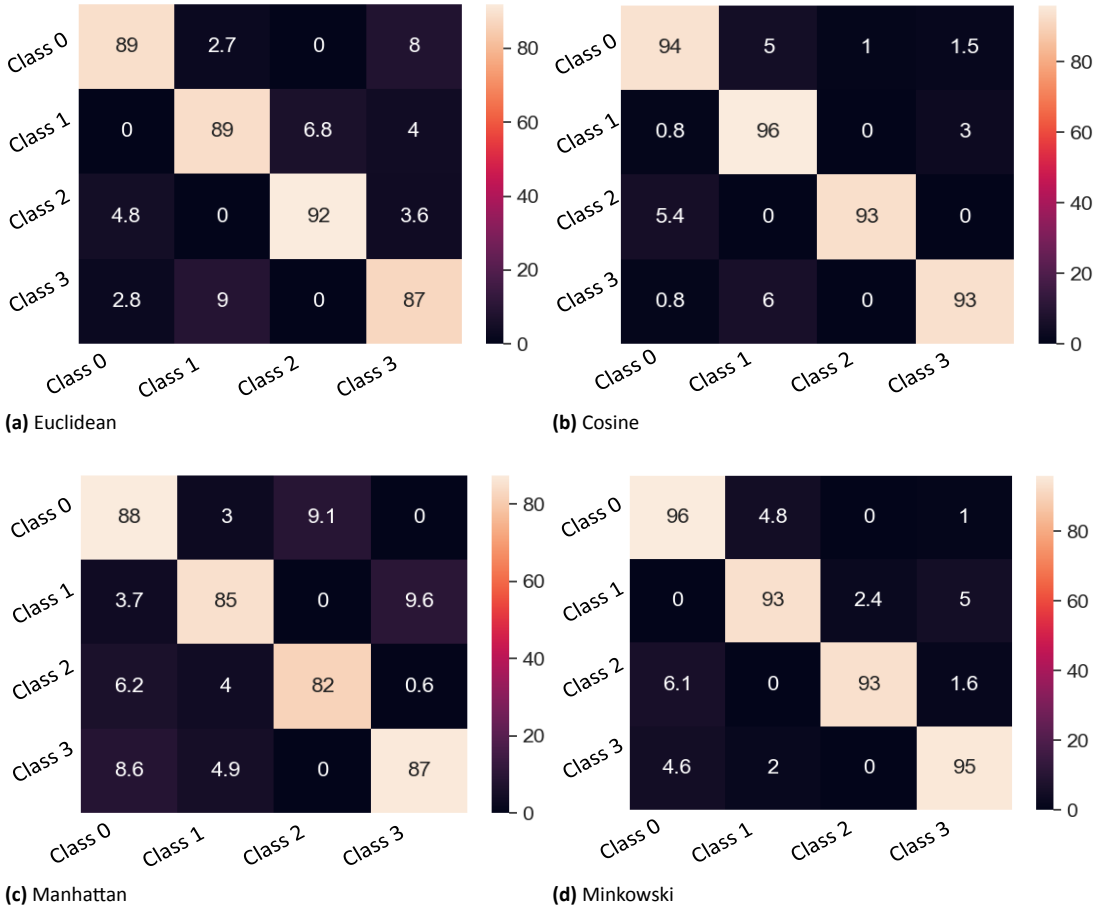
The results presented in Figure 5 in the form of scatter plots show the distribution of winning neurons for each training epoch. The data is based on training with Activity 1 only. Then, a set containing



(a) Euclidean **(b)** Cosine

(c) Manhattan **(d)** Minkowski

Figure 6. The scatter plots showing the distribution of the winning neurons for different distance metrics for Activity 12. Here $p1$ - person 1, and $p2$ - person 2, respectively



(a) Euclidean **(b)** Cosine

(c) Manhattan **(d)** Minkowski

Figure 7. The confusion matrix of proposed cNN taking into account the distance metrics for Activity 1

Table 2. Performance evaluation: cNN, KNN, and uRF with different distance metrics/configurations

Method	Distance	ACC	Pr	Re	F1-score
cNN	*Euclidean*	89.6%	91.3%	89.6%	90.0%
	Cosine	**95.6%**	**97.0%**	**95.2%**	**95.9%**
	Manhattan	87.7%	91.0%	87.7%	89.0%
	Minkowski	92.2%	94.6%	92.2%	94.0%
KNN [7]	*Euclidean*	73.9%	75.3%	73.1%	72.0%
	Cosine	**93.8%**	**94.1%**	**93.3%**	**93.5%**
	Manhattan	80.8%	82.1%	80.4%	80.3%
	Minkowski	82.2%	83.6%	81.2%	82.0%
uRF [15]	tree-16,0 depth-8	78.8%	79.3%	78.5%	78.6%
	tree-200, depth-10	78.2%	78.9%	78.2%	79.1%
	tree-220, depth-8	80.06%	**82.5%**	**81.9%**	80.1%
	tree-250, depth-8	**80.9%**	81.01%	80.65%	**80.9%**

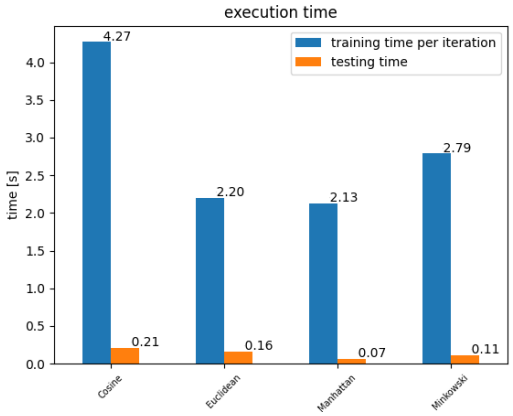


Figure 8. Computational efficiency of proposed cNN considering the distance metrics

the results of nine recording sessions for Activity 1 and Activity 12 was used for training. The number of outputs of the neural network was five. Scatterplots for this case are shown in Figure 6. In both examples, the separation of clusters is obvious, and the overlap of winning neurons is small, which justifies the validity of all considered metrics. Results are shown for the most significant *XY* (sagittal) plane.

In addition, Figure 7 shows the confusion matrix obtained when testing cNNs trained using only Activity 1. As seen, the results obtained for distances *Cosine* and *Minkowski* are better than those for *Euclidean* and *Manhattan* distances.

4. Experimental Evaluation

The final evaluation included the comparison of the proposed cNN with the results provided by the other commonly used networks with unsupervised training. For this evaluation, the data for all 20 activities were used. The number of cNN outputs was set to 20. The training was conducted using the data for nine randomly selected recording sessions for each activity, and the remaining three were used for testing.

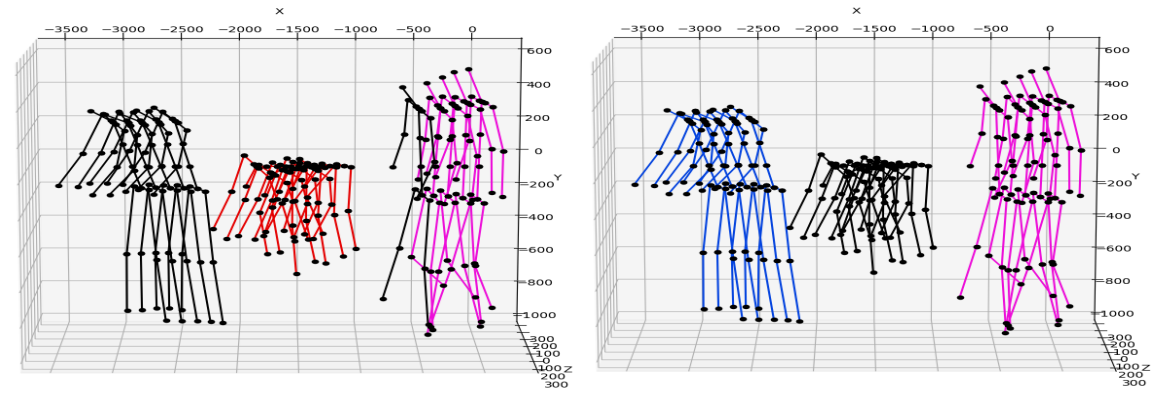
First, the performances of cNN, KNN, and uRF with four different distance metrics or configurations were compared. KNN is the Kohonen Neural Network [7], which belongs to a family of self-organizing feature maps that uses a competitive learning scheme with an additional parameter – the neighbourhood function. Therefore, the main parameters and structure of the KNN were similar to the proposed cNN (see the first paragraph of section 3.4). uRF is the unsupervised Random Forest [15]. According to [15], the uRF construction process involves modifying the Random Forest algorithm using k-means clustering. When building a modified Random Forest, the following parameters were used: (a) number of trees (n-estimators: 160, 200, 220, 250), (b) maximum depth (max-depth: 8, 10), (c) min-samples- split (8), (d) min-samples-leaf (16), (e) max-leaf-nodes (16), (f) max-features (sqrt), (g) criterion (entropy) respectively.

Referring to [9], a standard selection of assessment measures was used, namely accuracy (ACC), precision (Pr), memory (Re), and F1 score (F measure). A detailed description of the assessment measures can be found in [9]. Accuracy is most often used in classification problems. This measure, while intuitive, can be misleading in the case of a strongly unbalanced number of samples in [3] classes. Fortunately, this was not the case in the presented studies. Precision and recall are not very useful when considered separately because precision does not distinguish between true positives and true negatives, and recall does not inform how many cases were incorrectly classified. Taking into account the above, the F1-score was additionally used. It is the weighted harmonic mean of precision and recall (F1-score=2(Pr·Re)/(Pr+Re)). Testing results obtained using the above three methods and four different distance measures (if applicable), are given in Table 2. cNN with *Cosine* distance has a 95.6% accuracy.

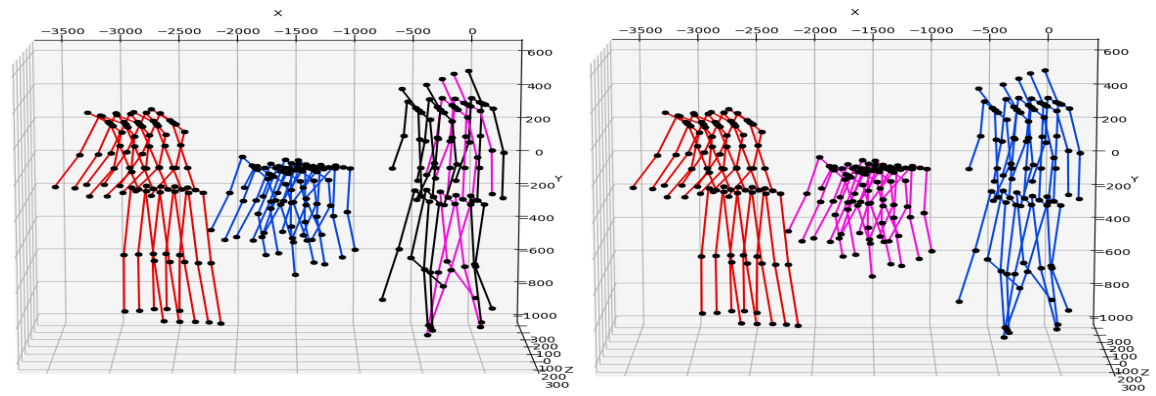
(a) frame 20 **(b)** frame 50 **(c)** frame 138

(d) frame 20 **(e)** frame 73 **(f)** frame 111

Figure 9. Sample image frames from both activity 1 and activity 12



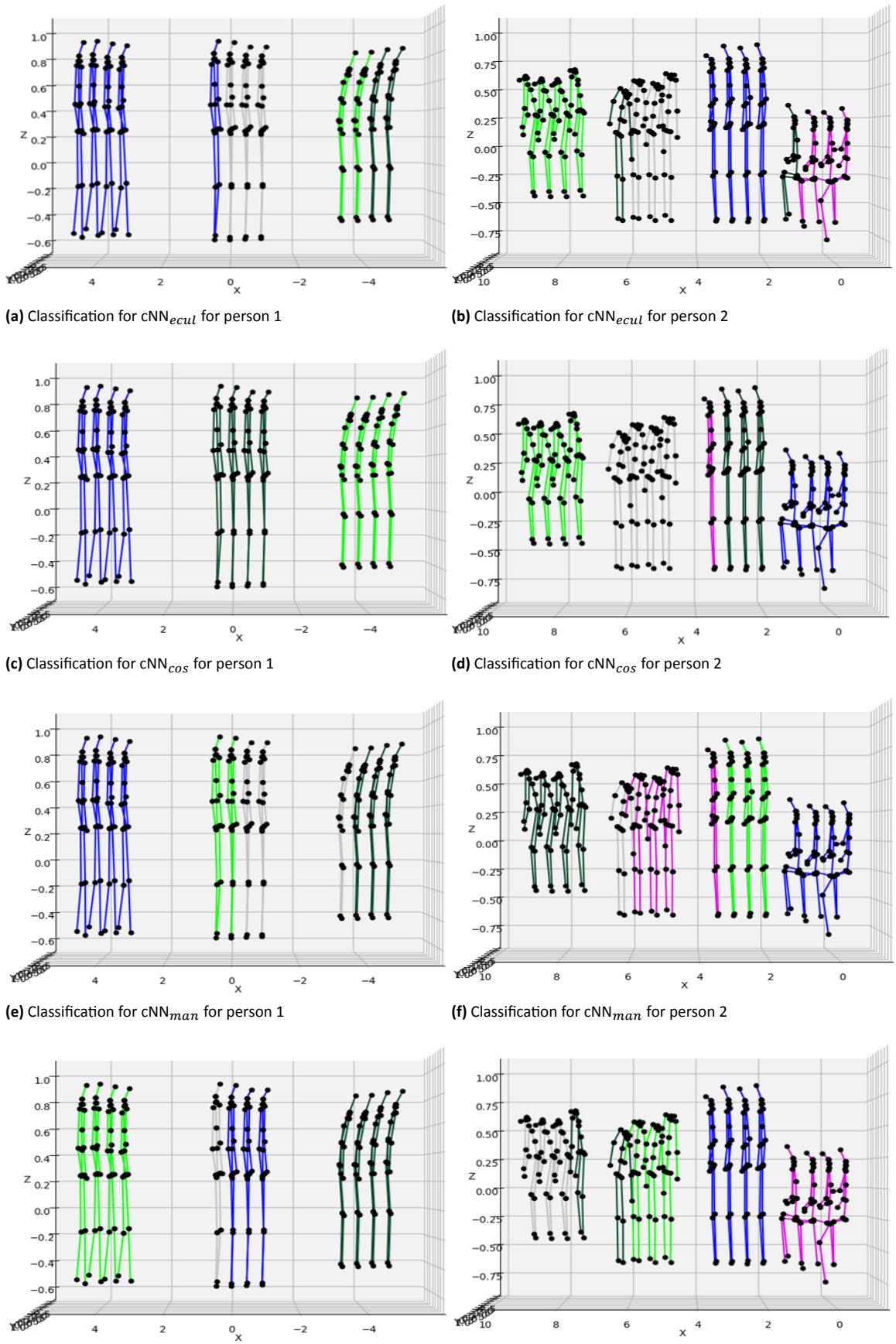
(a) Classification for cNN$_{ecul}$ **(b)** Classification for cNN$_{cos}$



(c) Classification for cNN$_{mah}$ **(d)** Classification for cNN$_{min}$

Figure 10. Stick diagrams of classification for Activity 1 (single person involved) are taken into account with four different distance metrics and configurations

A similar result concerns KNN with *Cosine* distance for which the accuracy is 93.8%. The classification quality is strong. The precision, recall, and F1 scores are high for all the above networks. Maximum scores obtained for each method are denoted by boldface font in Table 2. These studies led to the indication of the most appropriate distance metric. As Table 2 shows, the highest set of scores was obtained for cNN with *Cosine* distance. The second highest result was obtained for KNN with *Cosine* distance.



(a) Classification for cNN$_{ecul}$ for person 1

(b) Classification for cNN$_{ecul}$ for person 2

(c) Classification for cNN$_{cos}$ for person 1

(d) Classification for cNN$_{cos}$ for person 2

(e) Classification for cNN$_{man}$ for person 1

(f) Classification for cNN$_{man}$ for person 2

(g) Classification for cNN$_{min}$ for person 1

(h) Classification for cNN$_{min}$ for person 2

Figure 11. Stick diagrams of classification for Activity 12 (2 people involved) are taken into account with four different distance metrics and configurations

Next, the computational efficiency was examined to illustrate the cNN features better. Fig 8 shows the computational performance obtained for different distance metrics during training and testing. As can be seen in Figure 8, *Manhattan* distance gives the best results in terms of computational performance, followed by *Euclidean*. The *Cosine* distance metric shows the lowest efficiency.

Finally, a qualitative assessment was made by displaying the classified postures using stick diagrams of the human body. In this paper, the results obtained only for one testing session and for selected activities are presented. However, the results obtained for the remaining two sessions are similar, and the results for the remaining activities do not show excessive cases. Figure 9 shows example images for Activity 1 and Activity 12. The results illustrated by Figure 10 and Figure 11 confirm that the proposed cNN classifies the postures well. Each class is represented by one color in the stick diagrams.

cNN with distance *Cosine* and *Minkowski* correctly classified all postures for Activity 1. While testing using data for Acivity 12, cNN with distances *Cosine* and *Minkowski* correctly classified most postures. Only one posture was misclassified for person 1 when using the *Cosine* distance measure. Two misclassified plus one additional (unexpected by the expert) posture occurred in the case of *Minkowski* distance.

A single posture was misclassified for *Euclidean* distance, and even more errors occurred for *Manhattan* distance for Activity 1. On the other hand, for Activity 12, cNN with *Euclidean* distance had two misclassified positions for person 1, and even more for person 2. In the case of *Manhattan* distance, for both Activity 1 and Activity 12, several postures were assigned to unexpected or incorrect classes compared to the human expert indications (Fig. 10(c)).

5. Discussion

It should be emphasized that the data collection and processing method has been developed and verified. Further data collection is in progress. The studies described do not include data relevant to recognising human-human and human-object relationships. Such studies are planned for the future.

The described cNN network has been fully implemented, and the remaining neural networks have been used as ready-to-use programs. The software was developed in Python. Its libraries were also used, including machine learning tools, e.g., *scikit-learn* was used to evaluate the results, and *MatplotLib* was used for visualization. The calculations were performed on a PC with an Intel Core-i7 (2.8 GHz) processor and 64 GB of RAM.

Presented studies can be further extended. Only two RGB-D cameras collected the data, so using multi-camera data is a future research topic. Despite the use of many data sets, the effectiveness of the developed method of posture classification was verified experimentally, taking into account still a limited set of postures.

6. Conclusion

An improved version of a conventional cNN was programmed and tested with an adaptive selection of the most appropriate distance metric. Evaluations of the proposed approach made using the test dataset confirmed the good accuracy of the classification of the developed cNN network (95.6%) and KNN network (93.8%) with the *Cosine* distance measure.

It was also shown that the use of RGB-D sensors for human postures recognition is a good option wherever expensive professional motion capture systems are not available.

Applied type of neural network is classic. The network is simple and allows a full understanding of how it works. It is easy to modify so that it effectively meets the requirements. It was programmed from scratch because the ready-made libraries have many default functions that are not always required and may delay the work.

The proposed method allows for recognizing postures in real-time. The classification process was tested in a simulated online mode using recorded test data, and online classification was performed using a pre-trained neural network. The results were satisfactory, and the processing time was not noticeable to a human observer.

The indication that the angular measure is the most appropriate for the classification of postures is a valuable contribution. Changes in the angular positions of body parts cause changes in posture. In this sense, a larger number of output neurons means that postures with smaller angular differences will be grouped into one group, and fewer neurons mean that positions with more different postures are clustered together. By controlling the number of outputs neurons make, it is easy to influence the 'resolution' in postures recognition.

AUTHORS

Vibekananda Dutta* – Institute of Micromechanics and Photonics, Faculty of Mechatronics, Warsaw University of Technology, ul.Sw. Andrzeja Boboli 8, 02-525 Warsaw, Poland, e-mail: vibekananda.dutta@pw.edu.pl.

Jakub Cydejko – Institute of Automatics and Robotics, Faculty of Mechatronics, Warsaw University of Technology, ul.Sw. Andrzeja Boboli 8, 02-525 Warsaw, Poland, e-mail: jakub.cydejko2.stud@pw.edu.pl.

Teresa Zielińska – Institute of Aeronautics and Applied Mechanics, Faculty of Power and Aeronautical Engineering, Warsaw University of Technology, ul.Nowowiejska 24, 00-665 Warsaw, Poland, e-mail: teresa.zielinska@pw.edu.pl, www: https://ztmir.meil.pw.edu.pl/web/eng/Pracownicy/prof.-Teresa-Zielinska.

*Corresponding author

ACKNOWLEDGEMENTS

The research leading to the described results was carried out within the program POB-IDUB funded by Warsaw University of Technology within the Excellence Initiative Program – Research University (IDUB). The data used in this work are available from the corresponding author by request.

References

[1] A. R. Abas. "Adaptive competitive learning neural networks", *Egyptian Informatics Journal*, vol. 14, no. 3, 2013, 183–194.

[2] M. A. R. Ahad et al. "Action recognition using kinematics posture feature on 3d skeleton joint locations", *Pattern Recognition Letters*, vol. 145, 2021, 216–224.

[3] P. Branco, L. Torgo, and R. P. Ribeiro. "A survey of predictive modeling on imbalanced domains", *ACM Computing Surveys (CSUR)*, vol. 49, no. 2, 2016, 1–50.

[4] G. Budura, C. Botoca, and N. Miclău. "Competitive learning algorithms for data clustering", *Facta universitatis-series: Electronics and Energetics*, vol. 19, no. 2, 2006, 261–269.

[5] B. Cao, S. Bi, J. Zheng, and D. Yang. "Human posture recognition using skeleton and depth information". In: *2018 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*, vol. 1, 2018, 275–280, doi: 10.1109/WRC-SARA.2018.8584233.

[6] J. Chen, P. Jönsson, M. Tamura, Z. Gu, B. Matsushita, and L. Eklundh. "A simple method for reconstructing a high-quality ndvi time-series data set based on the savitzky–golay filter", *Remote Sensing of Environment*, vol. 91, no. 3-4, 2004, 332–344.

[7] K.-H. Chuang, M.-J. Chiu, C.-C. Lin, and J.-H. Chen. "Model-free functional mri analysis using kohonen clustering neural network and fuzzy c-means", *IEEE Transactions on Medical Imaging*, vol. 18, no. 12, 1999, 1117–1128.

[8] V. Dutta, and T. Zielinska. "An adversarial explainable artificial intelligence (xai) based approach for action forecasting", *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. 14, 2020.

[9] V. Dutta, and T. Zielinska. "Prognosing human activity using actions forecast and structured database", *IEEE Access*, vol. 8, 2020, 6098–6116.

[10] V. Farrahi, M. Niemelä, M. Kangas, R. Korpelainen, and T. Jämsä. "Calibration and validation of accelerometer-based activity monitors: A systematic review of machine-learning approaches", *Gait and Posture*, vol. 68, 2019, 285–299.

[11] M. Firman. "Rgbd datasets: Past, present and future". In: *IEEE Proc.*, vol. 1, 2016, 661–673.

[12] R. Hou et al. "Multi-channel network: Constructing efficient gcn baselines for skeleton-based action recognition", *Computers and Graphics*, vol. 110, 2023, 111–117.

[13] W. Kasprzak, and B. Jankowski. "Light-weight classification of human actions in video with skeleton-based features", *Electronics*, vol. 11, no. 14, 2022, 2145.

[14] V. Kellokumpu, M. Pietikäinen, and J. Heikkilä. "Human activity recognition using sequences of postures". In: *IAPR, Conference on Machine Vision Applications*, vol. 1, no. 1, 2022, 570–573.

[15] F. Kruber, J. Wurst, and M. Botsch. "An unsupervised random forest clustering technique for automatic traffic scenario categorization". In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, vol. 1, 2018, 2811–2818.

[16] J. Liu et al. "A graph attention spatio-temporal convolutional network for 3d human pose estimation in video". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, 2021, 3374–3380.

[17] L. E. Ortiz, V. E. Cabrera, and L. M. Goncalves. "Depth data error modeling of the zed 3d vision sensor from stereolabs", *ELCVIA: Electronic Letters on Computer Vision and Image Analysis*, vol. 17, no. 1, 2018, 1–15.

[18] E. J. Palomo, E. Domínguez, R. M. Luque, and J. Munoz. "A competitive neural network for intrusion detection systems". In: *International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences*, vol. 1, 2008, 530–537.

[19] G. Rogez, P. Weinzaepfel, and C. Schmid. "Lcr-net++: Multi-person 2d and 3d pose detection in natural images", *IEEE TPAMI*, vol. 42, no. 5, 2019, 1146–1161.

[20] L. Romeo, R. Marani, M. Malosio, A. G. Perri, and T. D'Orazio. "Performance analysis of body tracking with the microsoft azure kinect". In: *2021 29th Mediterranean Conference on Control and Automation (MED)*, vol. 1, 2021, 572–577, doi: 10.1109/MED51440.2021.9480177.

[21] D. E. Rumelhart, and D. Zipser. "Feature discovery by competitive learning". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1: Foundations*, 151–193. 1986.

[22] R. J. Saner, E. P. Washabaugh, and C. Krishnan. "Reliable sagittal plane kinematic gait assessments are feasible using low-cost webcam technology", *Gait and Posture*, vol. 56, 2017, 19–23.

[23] A. Shahroudy, J. Liu, T. Ng, and G. Wang. "Ntu rgb+d: A large scale dataset for 3d human activity analysis". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, Los Alamitos, CA, USA, 2016, 1010–1019, doi: 10.1109/CVPR.2016.115.

[24] C. SL. "Task compatibility of manipulator postures", *International Journal of Robotics Research*, vol. 7(5), 1988, 13–21, doi: 10.1177/027836498800700502.

[25] P. Tommasino, and D. Campolo. "An extended passive motion paradigm for human-like posture and movement planning in redundant manipulators", *Frontiers in Neurorobotics*, vol. 11, 2017, doi: 10.3389/fnbot.2017.00065.

[26] S. Vafadar, W. Skalli, A. Bonnet-Lebrun, M. Khalifé, M. Renaudin, A. Hamza, and L. Gajny. "A novel dataset and deep learning-based approach for marker-less motion capture during gait", *Gait and Posture*, vol. 86, 2021, 70–76.

[27] X. Wang, G. Liu, Y. Feng, W. Li, J. Niu, and Z. Gan. "Measurement method of human lower limb joint range of motion through human-machine interaction based on machine vision", *Frontiers in Neurorobotics*, vol. 15, 2021.

[28] L.-F. Yeung, Z. Yang, K. C.-C. Cheng, D. Du, and R. K.-Y. Tong. "Effects of camera viewing angles on tracking kinematic gait patterns using azure kinect, kinect v2 and orbbec astra pro v2", *Gait and Posture*, vol. 87, 2021, 19–26, doi: 10.1016/j.gaitpost.2021.04.005.

[29] T. Zielinska, G. R. R. Coba, and W. Ge. "Variable inverted pendulum applied to humanoid motion design", *Robotica*, vol. 39, no. 8, 2021, 1368–1389.