Neural network algorithm for predicting human speed based on computer vision and machine learning

Artem Obukhov[1*], *Daniil* Teselkin[1], *Ekaterina* Surkova[1], *Artem* Komissarov[2], and *Maxim* Shilcin[1]

[1]Department of Automated Systems of Decision-Making Support, Tambov State Technical University, Tambov, 392000, Russian Federation
[2] Department of Mechatronics and Technological Measurements, Tambov State Technical University, Tambov, 392000, Russian Federation

Abstract. The problem of increasing the accuracy of predicting human actions is an urgent task for various human-machine systems. The study examines the solution to the problem of predicting human speed using neural network algorithms, computer vision technologies, and machine learning. The formalization and software implementation of a neural network speed prediction algorithm are presented. To solve the problems of determining the current speed and predicting the upcoming positions of the human body depending on the dynamics of its movement, a comparison of various machine learning models was carried out. The RandomForestRegressor algorithm showed the best position prediction accuracy. The best determination of the current speed was demonstrated by dense multilayer neural networks. The experiment revealed that when predicting a person's position at an interval of 0.6 seconds, his speed is determined with an accuracy of more than 90%. The results obtained can be used to implement neural network algorithms for controlling human-machine systems.

1 Introduction

In human-machine systems, tracking user actions to generate relevant system responses is fundamental [1]. One of the modern ways to organize user tracking is computer vision technology, which has been rapidly developing in recent years due to the growth in the computing power of devices, the quality of cameras, and the emergence of more advanced software tools [2].

However, in the process of using human action tracking systems based on computer vision (or other motion capture technologies), a lag effect inevitably accumulates [3]. There are several factors that reduce system performance: time to receive data from a source (for example, cameras); time to transfer data to the processing terminal; data processing (object recognition using intelligent algorithms and machine learning models); transferring the result to the control system; formation of a command; response time of equipment or software tools

* Corresponding author: obuhov.art@gmail.com

to commands. As a result, the lag can last up to several seconds. For human-machine systems, such low performance leads to untimely responses to user actions. For example, when a person is already performing the next action or is not able to respond to an emergency situation in a timely manner [4], the system can send a control command that is no longer relevant. The main way to reduce lag is to predict user movements, which can be implemented by using various regression methods (ranging from conventional linear to more advanced machine learning methods).

The subject area of the study is determining the current speed and predicting the future speed of a person through the analysis of several consecutive frames. The use of various neural network architectures and machine learning algorithms is proposed as the main solution. As a result of the analysis, the lag will be reduced, which will ensure a timely response of the human-machine system to user actions.

2 Material and methods

The proposed neural network algorithm for predicting includes two main stages: determining the current speed of a person depending on the dynamics of his movement and forecasting variations in a person's velocity through the prediction of their future body positions.

In a formalized form, the following task is obtained: it is necessary to select a machine learning algorithm A and its parameters P, which approximate the regression relationship between a set of human body positions $\{X_{t-Q+1},...X_{t-1},X_t\}$ from Q number of measurements and its current speed s_t^* with a minimum error E relative to the real speed s_t:

$$\begin{gathered} A\left(\{X_{t-Q+1},...X_{t-1},X_t\},P\right)\to s_t^*, \\ E=\sum_{t=0}^{T}\sqrt{(s_t^*-s_t)^2}\to\min. \end{gathered} \tag{1}$$

Each element X_t contains the positions of key points, normalized relative to human body center.

To create a training dataset and collect the required amount of data, two methods are proposed:

- recording a person's movement on a video camera with subsequent capture of key points of the body; the person's speed is determined by setting a fixed speed value on the treadmill [5] and using manual start/stop to record changes in walking;
- simulation of the movement of a digital human model in a virtual environment, saving the animation in the form of a video sequence, and then capturing the position of all the necessary points and the current speed of the model's movement. This allows for varying the speed of movement by changing the animation speed [6].

The two methods are combined after obtaining information from two sources and normalizing the initial data on the position of the human body. The number of key points is determined by body recognition algorithms. Previous research shows that 18 points are enough to completely reconstruct a human body model [6]. To capture points of the body, it is possible to use various machine learning models (MediaPipe, MoveNet) and modern TokenPose models and their modifications [7]. Previous research shows that 18 points are enough to completely reconstruct a human body model [6].

Thus, to successfully solve problem (1), it is necessary to implement and compare various machine learning algorithms and neural networks. The analysis revealed that it is justified to use the following algorithms [8–10]:

- Regressors based on decision trees (DecisionTreeRegressor): easy to implement and interpret machine learning algorithm that allows for approximating the regression dependence;

- Regressors based on random forest (RandomForestRegressor): an ensemble classification method that combines several estimators (decision trees with a given depth) to improve accuracy;
- Multilayer dense neural networks: simple and universal approximators;
- LSTM type recurrent neural networks: common and effective models for time series analysis;
- Multilayer convolutional neural networks: designed to generalize the features of time sequences and series.

In addition to solving problem (1), it is necessary to predict the speed of a person at some point in time. Formalizing the stages of this task can be achieved by adapting the results of the previous study [11].

Let a set of data $H_t = \{X_{t-Q+1},...X_{t-1},X_t\}$ be given corresponding to the moment of time $t \in T$, which is denoted as the number of measurements Q of the object (user). According to the number of measurements Q, a prediction for W steps (W) can be made: $F_t = \{X_{t+1},...X_{t-1+W},X_{t+W}\}$. The lengths of Q and W may be equal or vary. However, it is worth taking into account that a larger Q simplifies predicting due to an increase in the amount of data, while a larger W reduces the prediction accuracy.

The process of training a neural network can be formulated as follows: for each output vector j in the training sample $Y_j^t = (X_{j+1},...,X_{j+W-1},X_{j+W})$, the corresponding input vector $X_j^t = (X_{j-Q+1},...,X_{j-1},X_j)$ is composed of Q previous sets of the object's states. Thus, the input of the machine learning algorithm A_F is composed of Q previous states of the object, and the output consists of W subsequent states. Then the optimization task of the algorithm A_F is to minimize the standard error E of the predicted values from their real values:

$$A_F : X^t \to Y^t,$$
$$E(A_F) \to \min. \tag{2}$$

To solve problem (2), various machine learning algorithms discussed above are also used, since their architecture involves solving both problems by changing only the output layer.

Figure 1 shows a scheme of the experiment reflecting the progress of solving problems 1 and 2.

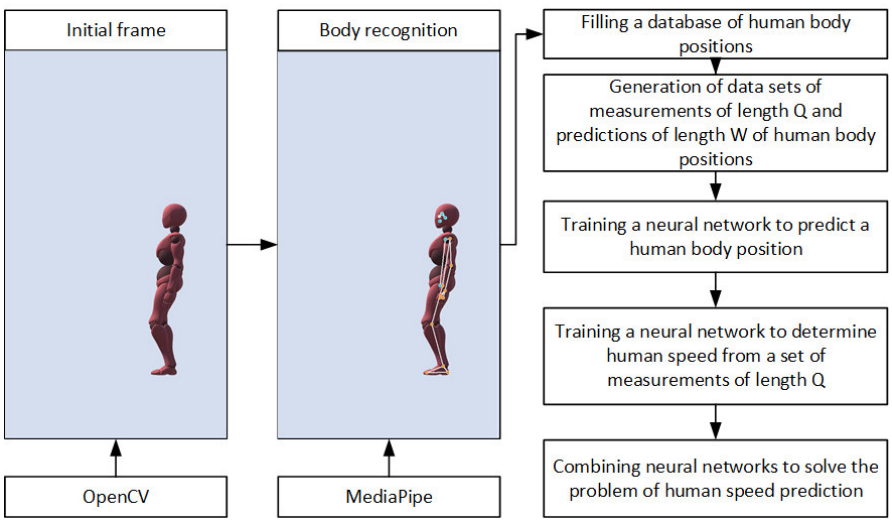

Fig. 1. Scheme of the neural network predicting algorithm.

At first, the original frame from the camera is converted into a set of coordinates of human body key points. Next, it is necessary to collect data about the movement and generate Q number of key point positions of the human body, which are fed to the input of the neural network model. This makes it possible to reduce the lag of the control system by $W(1/FPS)$ ms, where FPS is the frequency of the video sequence received from the camera.

3 Results and discussion

To solve problems (1) and (2), a comparison of various machine learning algorithms and neural network models was carried out. A summary of the final optimal characteristics of algorithms and models is presented in Table 1.

Table 1. Final characteristics of machine learning algorithms.

Model	Description
DecisionTreeRegressor (DTR)	Standard decision tree regressor with parameter max_depth = 5
LinearRegression (LR)	Ordinary least squares linear regression with default parameters
RandomForestRegressor (RFR)	Standard random forest regressor with parameters n_estimators = 10, max_depth = 5
Convolutional neural network (CNN)	Multilayer convolutional network of 4 pairs of Conv1D and BatchNormalization layers (number of filters: from 32 to 256), GlobalAvgPool2D, and Dense layers
Recurrent neural network (LSTM)	Multilayer neural network with 2 hidden Dense layers of 100 and 200 neurons and 2 LSTM layers of 20 neurons
Simple neural network (NN)	Multilayer neural network with an input consisting of a hidden Dense layer of 200 neurons with a ReLU activation function and a Dropout layer with 20% pruning

Each of the architectures was used to solve problems 1 and 2. The results for the control sample not involved in training or testing the algorithms are presented in Table 2. For the first task, 4 tests were carried out, two of which did not use predicted data ($W = 0$). In addition, the size of Q and W varied: 10 and 15 frames were used, which corresponds to 0.4 and 0.6 seconds. The total volume of collected data on human movement at different speeds on an active running platform was 3495 records for training and testing and 548 records for control validation. To assess the quality of the models, the metric of mean absolute error (MAE) between the actual speed and the prediction was used. For the second task, the deviation for each tracked point of the human body model was calculated, and then the MAE was determined.

Table 2. Comparison of machine learning algorithms using the MAE metric.

Model	Speed determination problem (1)				Speed prediction problem (2)	
	Q=10, W=0	**Q=10, W=10**	**Q=15, W=0**	**Q=15, W=15**	**Q=10, W=10**	**Q=15, W=15**
DTR	0.101	0.213	0.095	0.2	0.0033	0.0037
LR	0.496	0.489	0.489	-	0.0034	0.0035
RFR	0.1	0.158	0.096	0.163	**0.0021**	**0.0022**
CNN	0.164	0.183	0.163	0.177	0.0101	0.0103
LSTM	0.196	0.216	0.171	0.198	0.0064	0.0069
NN	**0.084**	**0.133**	**0.081**	**0.129**	0.0044	0.0048

In the first problem, it was found that when determining the speed based only on the current position values for 10 or 15 frames, the NN model shows the best performance. Analysis of predictions for 10 and 15 frames shows that the NN model also has the smallest error. The second most accurate model is RFR. The LR model at Q=15, W=15 was unable to correctly approximate the speed, showing too high an error, which makes this model unsuitable for use.

The solution to problem (2) is provided by almost all algorithms; the most effective is the RFR model. The result of applying the RFR model to predict position and NN to determine speed is presented in Figure 2.

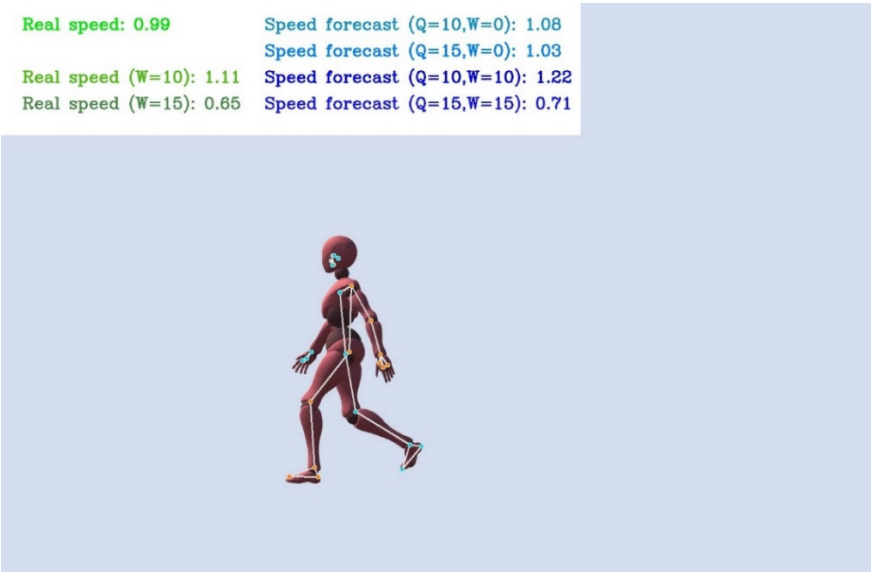

Fig. 2. The use of trained models to predict speed.

This experiment uses video obtained by capturing a virtual scene in which a digital human model is walking. There is a certain increase in error when using a forecast of 10 or 15 frames, but this allows for a reduction in lag by 0.4 or 0.6 seconds, respectively. By assessing the error during movement (at a speed of 1 m/s), a fairly high prediction accuracy was obtained. The possible braking of a person during walking is also taken into account (in Figure 2, the brake occurred after 15 frames and was predicted quite accurately). Thus, the accuracy of speed prediction using RFR and NN ranged from 91-96% (at W=0) to 90.8% (at W=15), which can be considered an acceptable result. The obtained result concludes that neural network forecasting algorithms can determine a person's speed with over 90% accuracy for a prediction interval of up to 0.6 seconds.

4 Conclusion

The study examines the problem of predicting the human movement speed using neural network algorithms. The solution to this problem can be used in applied research, for example, when implementing control algorithms for active running platforms. Information about the position of points in the human body obtained using computer vision technologies is used as initial data.

As a result of the research, a neural network prediction algorithm was formalized, including the solution to two problems: determining the position of a person through W

number of frames and determining his speed using Q number of frames. As a result of solving these problems, the lag effect of the control system was reduced. During the research, data on human movements was collected using computer vision and the MediaPipe library, after which various neural network models and machine learning algorithms were compared. The RandomForestRegressor algorithm showed the best accuracy when predicting position and multilayer neural networks when determining speed. It was found that when predicting a person's position in an interval of 15 frames (approximately 0.6 seconds), the person's speed is determined with an accuracy of more than 90%. The results obtained can be used to implement neural network algorithms for controlling systems with human presence, for example, in treadmills. The direction of further research is to increase the interval of analysis and prediction of human movements, expand the learning dataset, and increase the accuracy of the neural network algorithm.

The research was carried out at the expense of the grant of the Russian Science Foundation No. 22-71-10057, https://rscf.ru/en/project/22-71-10057/.

References

1. R. Yin, D. Wang, S. Zhao, Z. Lou, G. Shen, Adv. Funct. Mater. **31**, 2008936 (2021)
2. B.F. Spencer Jr, V. Hoskere, Y. Narazaki, Engineering-London **5**, 199-222 (2019)
3. H. Kwon, C. Tong, H. Haresamudram, Y. Gao, G.D. Abowd, N.D. Lane, T. Ploetz, Proc. ACM interact. mob. wearable ubiquitous technol **4**, 1-29 (2020)
4. R. Nabiei, M. Najafian, M. Parekh, P. Jančovič, M. Russell, *Delay reduction in real-time recognition of human activity for stroke rehabilitation* in 2016 First International Workshop on Sensing, Processing and Learning for Intelligent Machines, SPLINE, 6-8 July 2016, Aalborg, Denmark (2016)
5. A. Obukhov, A. Nazarova, K. Patutin, E. Surkova, D. Teselkin, *Development of Software for Managing Treadmills Based on Computer Vision* in International Conference Artificial Intelligence in Models, Methods and Applications, AIES-2022, 15-18 November 2022, remote (2022)
6. A. Obukhov, D. Dedov, A. Volkov, D. Teselkin, Comput. **11**, 85 (2023)
7. Y. Li, S. Zhang, Z. Wang, S. Yang, W. Yang, S.T. Xia, E. Zhou, *Tokenpose: Learning keypoint tokens for human pose estimation* in Proceedings of the IEEE/CVF International conference on computer vision, ICCVW, 11-17 October 2021, Montreal, Canada (2021)
8. D. Yacchirema, J.S. de Puga, C. Palau, M. Esteve, Computing **23**, 801-817 (2019)
9. K. Bartol, D. Bojanić, T. Petković, S. Peharec, T. Pribanić, Sensors **22**, 1885 (2022)
10. S. Turgeon, M.J. Lanovaz, Perspect. Behav. Sci. **43**, 697-723 (2020)
11. A.D. Obukhov, M.N. Krasnyanskiy, Neural. Comput. Appl. **33**, 15457-15479 (2021)