ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

Using LSTM Recurrent Neural Networks to Predict the Trajectory of Human Hand Movement in the Working Area of a Collaborative Robot-Manipulator

Svitlana Maksymova [1], Ahmad Alkhalaileh [2], Dmytro Gurin [1], Vladyslav Yevsieiev [1]

[1] Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv National University of Radio Electronics, Ukraine

[2] Senior Developer Electronic Health Solution, Amman, Jordan

Abstract: The article examines the use of LSTM recurrent neural networks for predicting the trajectory of human hand movement in the working area of a collaborative robot-manipulator. The results demonstrate high prediction accuracy for slow movements, but reveal certain limitations for fast and complex trajectories. The proposed approach is aimed at improving the safety and efficiency of the joint work of humans and robots within the framework of the concept of Industry 5.0.

Key words: Industry 5.0, Collaborative Robot, Work Area, Computer Vision, LSTM, Trajectory Prediction.

Introduction

In today's world, when the concepts of Industry 5.0 are becoming more and more relevant, the interaction between humans and robots takes on a new meaning [1]-[21]. Industry 5.0 emphasizes the harmonization of relations between man and machine, where robots play the role of intelligent assistants, enhancing human capabilities without interfering with his creative potential. One of the important aspects of such cooperation is ensuring the safety and efficiency of human work in the working area of the collaborative robot-manipulator [22]-[26]. Various appropriate methods and approaches can be used here [27]-[43]. Predicting the trajectory of the operator's hands is of key importance to prevent possible collisions and abnormal situations that may arise in the process of joint work. In this context, the use of recurrent neural networks (RNN), in particular Long Short-Term Memory (LSTM) networks, becomes extremely relevant. LSTM networks are able to efficiently analyze sequential data, detect long-term dependencies, and predict future events based on historical data. This makes it possible to accurately predict the trajectory of human hand movement in the robot's

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

working area, which is extremely important for ensuring dynamic safety and adaptive control of a collaborative robot.

The relevance of this study is due to the rapid development of cyber-physical systems, where a person and a robot work side by side in a single production process. Security and interaction in such an environment require new approaches to analyzing and predicting both human and robot behavior. The use of LSTM networks for this task allows to achieve a higher level of integration and synergy between man and machine, which corresponds to the principles of Industry 5.0. Therefore, the study of methods of predicting the trajectory of the operator's hands using LSTM is not only technically interesting, but also extremely important from the point of view of safety, efficiency and development of modern production processes.

Related works

The extreme relevance of using collaborative robots poses new challenges for scientists and developers. Accordingly, an ever-increasing number of works appear devoted to solving various problems that arise during the joint work of humans and robots. Let us briefly consider some of these works.

Collaborative robots are innovative industrial technologies introduced to help operators to perform manual activities in so called cyber-physical production systems and combine human inimitable abilities with smart machines strengths. Occupational health and safety criteria are of crucial importance in the implementation of collaborative robotics [44].

Let us begin with the work [45] that is intended to delineate an interpretation key for the design of collaborative robotics solution that explains the relationship among all relevant factors: actuation, control, safety, physical interaction, usability, and productivity.

Scientists in [46] also review the significance of the collaborative robots today and also present an insight into their future potential.

The article [47] reviews the development of cobots in manufacturing and discusses future opportunities and directions from cobots and manufacturing system perspectives in order to incentive future research. It provides novel and valuable insights into cobots application and illustrates potential developments of future human-cobot interaction.

It is important to recognize that risk assessment remains a crucial tool for safety with both collaborative and non-collaborative industrial robot systems [48]. The nature

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

of collaborative robots is discussed in [48] and how they are currently being used in industry; voluntary industry consensus standards for safety of collaborative robot applications; and best practices in evaluating and mitigating the new hazards related to collaborative robot applications.

Knudsen, M., & Kaivo-Oja, J. in [49] provide novel and valuable insights. In highlighting current frontiers, they also illustrate potential developments of future human-robot interaction.

Researchers in [50] note, that cobots need additional mechanisms to assure humans' safety in collaborations. In [50] the needs of the safety assurance of integrated robotic systems are specially discussed with two development examples.

The study [51] reviews requirements for safety assurance of collaborative robot systems discussed in the recent ISO 15066 standard for collaborative robots and how such safeguards are realized in studies discussed in literature. The review [51] explores gaps and propose a framework based on the ISO 31000 for orienting design safeguards for collaborative robots to outcomes of hazard analysis and risk assessment.

Thus, we see that many works are devoted to the development of collaborative robots. Later in this article, we will consider a computer vision system for a collaborative robot that is capable of predicting the movements of human hands in the working area of such a robot.

Mathematical presentation of the principle of predicting the trajectory of human hand movement based on the LSTM model in the working area of a collaborative manipulator robot

The LSTM (Long Short-Term Memory) model is used to predict the trajectory of human hand movement in the working area of a collaborative robot-manipulator due to its ability to take into account time dependencies in sequential data. The basic principle of LSTM is that it is able to remember important information from previous steps and use it to make decisions in subsequent steps. This is achieved thanks to a special architecture that includes memory cells that allow you to store or discard information through the mechanisms of "input" and "forgetting" gates. When the LSTM receives a sequence of hand movement coordinates, it analyzes them, identifying patterns and dependencies between previous and current positions. So, the model can predict where the hand will move at the next moment in time. In the context of a collaborative robot, this allows the robot to dynamically adapt its actions, avoiding collisions and ensuring operator safety.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

Based on this, in the framework of these studies, the input data is a sequence of coordinates of a point on the hand, which are obtained thanks to the use of graph theories in the last *t* frames. Then the input data can be rearranged as follows:

$$X=\{(x_1,y_1), (x_2,y_2), \ldots, (x_t,y_t)\} \quad (1)$$

x_t and y_t - normalized coordinates of the object on the frame *i*;

The LSTM (Long Short-Term Memory) model is used to process and predict time series where the sequence of input data is of great importance. In the context of the motion trajectory prediction task, each parameter and input vector of the LSTM structure has a purpose. As a result, the LSTM model accepts as input a sequence *X* of length *t* and generates a prediction of object coordinates for the next step (frame):

$$X=\{(x_t,y_t), (x_{t-1},y_{t-1}), \ldots, (x_{t-n+1},y_{t-n+1})\} \quad (2)$$

x_t,y_t - coordinates of the object at the last time step *t*;

x_{t-1},y_{t-1} - coordinates of the object at the previous time step *t-1*;

x_{t-n+1},y_{t-n+1} - coordinates of the object at the last time step *t-n+1*.

Each LSTM block consists of several important components: inputs, memory cells, and several gates that determine how information will be processed at each time step.

The input gate determines how much new information should be added to the memory state:

$$i_t=\sigma(W_{ix}X_t+W_{ih}X_{t-1}+b_i) \quad (3)$$

i_t - the output of the input gate at the time step *t* etermines how much of the new information (arrived at the current time step) should be added to the memory state. i_t has a value between 0 and 1, where 0 means that no information is added and 1 means that information is completely added to the memory state;

σ - sigmoid activation function. It is applied to a linear combination of the input data and the previous state to obtain a value between 0 and 1, and determines the probability of how much new information should be added to the memory state;

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

W_i - the weight matrix for the input gate, which is multiplied by X_t the input data at the time step t, and determines the weight with which new information (input data) will affect the input gate;

X_t - the vector of input data at the time step t contains the information that we provide to the input of the LSTM model (for example, object coordinates or other attributes);

W_{ih} - the weight matrix for the input gate, which is multiplied by the hidden state vector h_{t-1} from the previous time step $t-1$;

h_{t-1} - the hidden state vector at the previous time step $t-1$ contains information that has been preserved from previous time steps and is used to make a decision on updating the memory state

b_i - bias for the input gate. A bias is added to the linear combination of the input data and the hidden state to provide additional model flexibility.

The forgetting gate determines how much of the previous information needs to be "forgotten" and can be described by the following expression:

$$f_t = \sigma(W_{fx}X_t + W_{fh}X_{t-1} + b_f) \tag{4}$$

The output gate, which determines what part of the current memory state should be transferred to the output, can be described as follows:

$$o_t = \sigma(W_{ox}X_t + W_{oh}X_{t-1} + b_o) \tag{5}$$

The memory state (C_t) is updated at each step as follows:

$$C_t = f_t \otimes C_{t-1} + i_t \otimes HTan(W_{cx}X_t + W_{ch}X_{t-1} + b_c) \tag{6}$$

f_t - forgetting gate on time step t, it determines how much of the previous memory state C_{t-1} should be kept in the new memory state;

C_{t-1} - the previous state of the memory at the time step $t-1$, it stores the long-term information that the model has accumulated up to this point;

i_t - the input gate at the time step t, it controls how much of the new information that arrived at the time step t should be added to the memory state C_t;

HTan - hyperbolic tangent activation function. It transforms a linear combination of the input data and the hidden state into a value between -1 and 1;

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

W_{cx} - a matrix of weights for new information that is multiplied by X_t the input at a time step t;

X_t - the vector of input data at the time step t, contains information that we provide to the input of the LSTM model (for example, object coordinates or other attributes);

W_{ch} - the weight matrix for the new information, which is multiplied by h_{t-1} the hidden state from the previous time step $t-1$;

b_c - bias for new information, it is added to the linear combination of the input data and the hidden state to provide additional model flexibility.

Expression 6 defines how the memory state is updated in the LSTM model at each time step. It combines the previous state of the memory with new information, taking into account what part of the previous state should be preserved (through the forgetting gate f_t) and what part of the new information should be added (through the input gate i_t).

Updates of the hidden state (h_t) can be described as follows:

$$h_t = o_t \otimes HTan(C_t) \qquad (7)$$

h_t - the hidden state at a time step t, contains short-term information that will be used as an output at this time step and transmitted to the next time step;

o_t - the output gate at time step t, determines which part of the memory state C_t should be output as a hidden state h_t;

$HTan$ - hyperbolic tangent activation function, it normalizes the memory state C_t, limiting its value between -1 and 1;

C_t - the memory state at time step t, stores the information that the model has accumulated up to this moment of time.

Expression 7 defines how the output hidden state h_t is formed based on the current memory state C_t. The output gate o_t controls how much information from C_t is passed to the output hidden state, which affects the short-term dynamics of the model and how information is passed to subsequent time steps or to the output layer of the model.

The output forecast is obtained on the last layer of the LSTM model, which predicts the coordinates of the object, and can be presented as follows:

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

$$\tilde{Y}_{t+1} = W_{hy} h_t + b_y \qquad (8)$$

$\tilde{Y}_{t+1}$ - predicted output value at the next time step *t+1*, this value is the result of the model, used to estimate the future state of the system or a certain indicator;

W_{hy} - the weight matrix coefficient between the hidden state h_t and the original forecast $\tilde{Y}_{t+1}$;

h_t - the hidden state at time step t, contains short-term information that is used to predict the next value $\tilde{Y}_{t+1}$;

b_y – shift vector (bias) for the output $\tilde{Y}_{t+1}$, is added to a linear combination of weights and latent state to adjust the model, helping it to better match the predicted values with the actual values.

Expression 8 defines the process of predicting the output value $\tilde{Y}_{t+1}$ based on the hidden state h_t, which was calculated at the previous time step. The weight matrix coefficient W_{hy} and shift vector (bias) b_y adjust for the influence of the hidden state on the prediction, allowing the model to learn relevant dependencies in the data and use them to make accurate predictions.

Within the framework of these studies, we will give the following definition that the predicted trajectory is a set of consecutive predicted points generated on the basis of current information and previous forecasts:

$$T = \{(\tilde{x}_{t+1}, \tilde{y}_{t+1}), (\tilde{x}_{t+2}, \tilde{y}_{t+2}), \ldots, (\tilde{x}_{t+k}, \tilde{y}_{t+k})\} \qquad (9)$$

$\tilde{x}_{t+1}$, $\tilde{y}_{t+1}$ - predicted coordinates of a point on a plane (for example, on a screen or in space) at the next time step *t+1*, they determine the position of an object (for example, a point on a hand or a face) one step forward in time;

$\tilde{x}_{t+2}$, $\tilde{y}_{t+2}$ - predicted coordinates of a point on a plane (for example, on a screen or in space) at the next time step *t+2*, they determine the position of an object (for example, a point on a hand or a face) at a second step forward in time;

$\tilde{x}_{t+k}$, $\tilde{y}_{t+k}$ - predicted coordinates of a point on a plane (for example, on a screen or in space) at a time step *t+k*, they determine the position of an object (for example, a point on a hand or a face) at the *k*-th step forward in time

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

According to the fact that we receive data from the camera, in expression 9 it is necessary to take into account scaling to the screen size, as a result, expression 9 will have the following form:

$$T_{scaled} = \{(\tilde{x}_{t+1} \times W, \tilde{y}_{t+1} \times H), ..., (\tilde{x}_{t+k} \times W, \tilde{y}_{t+k} \times H)\} \quad (10)$$

W and H - width and height of the window, respectively.

So, in accordance with the above, the mathematical description of the predicted trajectory based on the LSTM model includes the processing of input data using LSTM layers, which take into account both current and past information, and generate the predicted coordinates of the object. These coordinates can be used to visualize the trajectory of movement in space. This approach allows you to take into account time dependencies and provide predictions of future positions of the object based on its history of movement within the framework of collaborative work with the robots-manipulators within the framework of Industry 5.0 concepts.

Software implementation of the LSTM model for predicting the trajectory of the movement of human hands in the working area of a collaborative robots-manipulator

Python is an ideal choice for a software implementation of an LSTM model for predicting human hand movement trajectories in the workspace of a collaborative robot-manipulator for several reasons. First, Python has a rich set of libraries, such as TensorFlow and Keras, that provide simple and efficient implementations of complex neural networks, particularly LSTMs. Second, Python supports numerous data processing and visualization tools, such as NumPy, Pandas, and Matplotlib, which facilitate model analysis and debugging. In addition, Python is widely used in machine learning and robotics, making it popular among researchers and developers. Its simple syntactic structure facilitates rapid prototyping and reduces development time, which is critical for rapid iteration and improvement of models in a dynamic robotics environment.

We will give an example of the implementation of some functions in the developed program for predicting the trajectory of the movement of human hands in the working area of a collaborative robot-manipulator based on the LSTM model.

```
mp_hands = mp.solutions.hands
mp_drawing = mp.solutions.drawing_utils
```

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

This piece of code is for configuring the MediaPipe library to detect hands in an image or video. mp_hands initializes the module responsible for recognizing and tracking key points of the hand, and mp_drawing provides the tools to visualize and draw these points on an image or video. This is necessary for further analysis and display of the position of the hands in the frame.

```
model = Sequential([
    LSTM(50, activation='relu', input_shape=(sequence_length, 2), return_sequences=True),
    LSTM(50, activation='relu'),
    Dense(2)
])
model.compile(optimizer='adam', loss='mse')
```

This piece of code creates and configures an LSTM model to predict hand movement trajectories. It contains two LSTM layers to process the data sequence and one Dense layer to generate the final coordinates. The model is compiled using the `adam' optimizer and the `mse' (mean squared error) loss function.

```
ih, iw, _ = frame.shape
            x, y = hand_landmarks.landmark[8].x, hand_landmarks.landmark[8].y
            cv2.circle(frame, (int(x * iw), int(y * ih)), 5, (0, 255, 0), -1)
```

This piece of code determines the coordinates of the tip of the index finger in the image and displays it as a green circle in the video. It uses the frame size to scale the point coordinates according to the video resolution.

```
predicted = model.predict(input_sequence)[0]
            trajectory = np.array([hand_landmarks_history[i] for i in range(-sequence_length, 0)] + [predicted])
```

This piece of code uses an LSTM model to predict the next hand position based on the current sequence of coordinates. The result is predicted and added to the history of the coordinates to construct the hand movement trajectory.

```
            for j in range(len(trajectory) - 1):
                start_point = (int(trajectory[j][0] * iw), int(trajectory[j][1] * ih))
                end_point = (int(trajectory[j + 1][0] * iw), int(trajectory[j + 1][1] * ih))
                cv2.line(frame, start_point, end_point, (0, 0, 255), 2)
```

JOURNAL OF UNIVERSAL SCIENCE RESEARCH
ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

This piece of code draws a line on the image connecting successive points of the trajectory of the predicted hand movement. It uses the coordinates of the points to visualize the predicted trajectory as a red line on the video.

The results of the program for predicting the trajectory of the movement of human hands in the working area of the collaborative robots-manipulator, through the computer vision system, are shown in Figure 1.

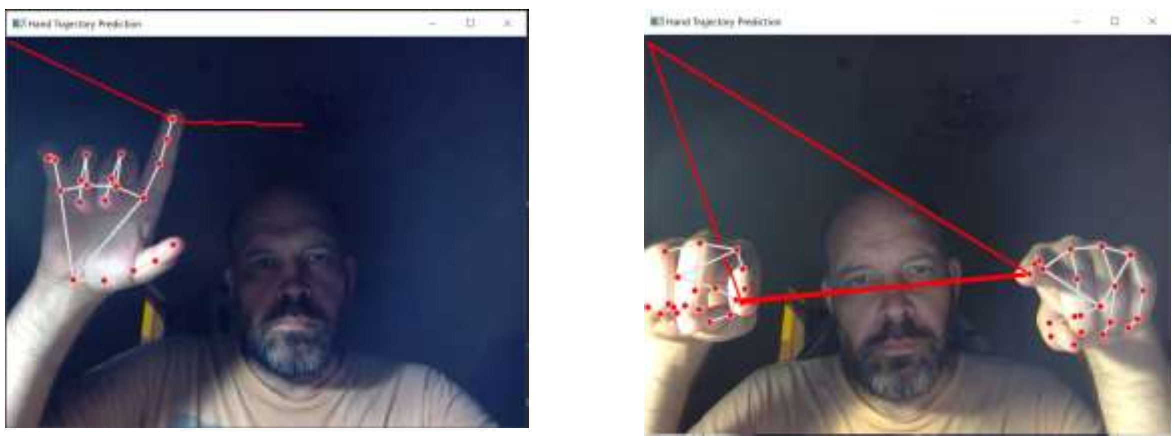

Figure 1: Results of the program for predicting the trajectory of the movement of human hands in the working area of a collaborative robots-manipulator.

Based on the developed program, a number of experiments were conducted on the accuracy of trajectory prediction and with different types of hand movements (fast, slow, complex). For the purity of the experiment, we note that the hardware consisted of the following elements: CPU Intel Core i7-6650U, 3.4 GHz; RAM – 16Mb; HDD - 512Gb; GPU Intel Iris Graphics 540. The obtained results are presented in Table 1, and for the convenience of visualization for data analysis are presented in the form of graphs in Figure 2.

Table 1 - Results of the experiment on the accuracy of trajectory prediction and with different types of hand movements (fast, slow, complex).

Type of movement	Average prediction error (pixels)	Average movement speed (pixels/sec)	Tests number	Forecast accuracy (%)
straight line (slow)	5.3	10		94.7
straight line (fast)	12.8	50		87.2

JOURNAL OF UNIVERSAL SCIENCE RESEARCH

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

curvilinear (slow)	8.6	15	10	91.4
curvilinear (fast)	15.2	45		84.8
random (slow)	10.1	12		89.9
random (quick)	18.3	55		81.7
complex (slow)	9.7	20		90.3
complex (quick)	16.5	48		83.5

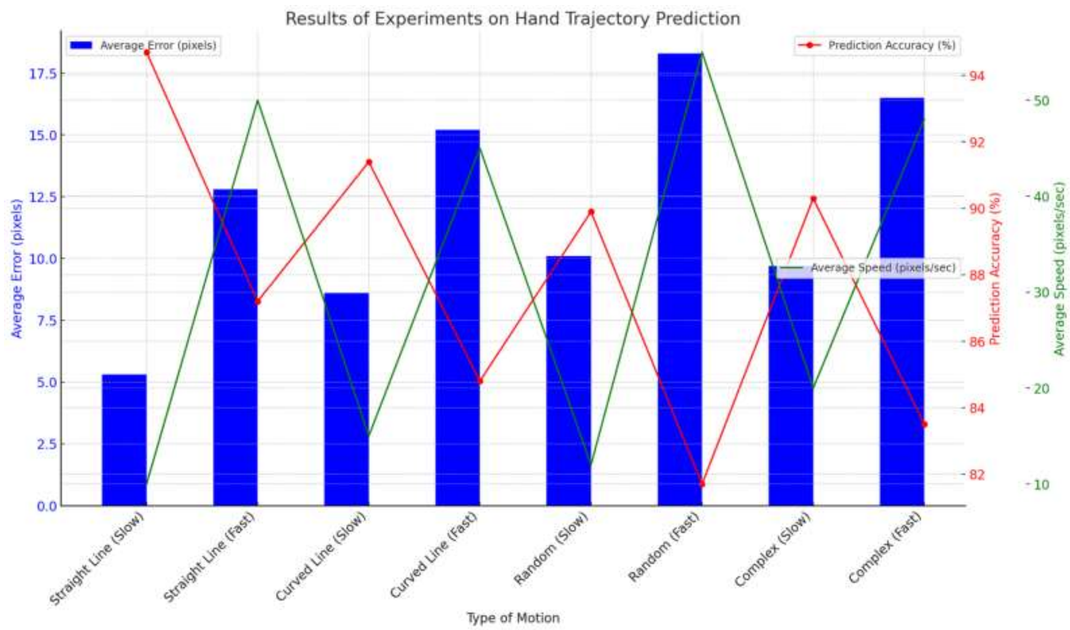

Figure 2: Graph of the obtained results of the experiment on the accuracy of trajectory prediction and with different types of hand movements (fast, slow, complex) using LSTM neural networks

Figure 2 shows:

- average forecast error (Average Error) in the form of blue bars;
- prediction accuracy (Prediction Accuracy) with a red line;
- the average speed of movement (Average Speed) with a green line.

The conducted experiments show that the accuracy of hand movement trajectory prediction using LSTM depends significantly on the type of movement and its speed. Slow movements, regardless of their complexity, show higher prediction accuracy and lower average error, confirming the effectiveness of LSTM for predictable and stable trajectories. Fast movements, on the contrary, are characterized by a significant increase in error and a decrease in accuracy, which may be due to the limited capabilities of the

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

model to process rapid changes in input data. Complex trajectories, even with slow motion, also show slightly higher error compared to simple motions, which may indicate the need for additional training of the model to handle more parameters. The obtained results indicate the importance of adapting the model to the specific conditions of the task, which may include the optimization of LSTM parameters or the use of additional algorithms to increase the accuracy of prediction in conditions of complex and fast movements.

Conclusion

The research findings show that the use of LSTM recurrent neural networks to predict the trajectory of human hand movement in the working area of a collaborative robots-manipulator is a promising approach that demonstrates high accuracy under conditions of slow and predictable movements. The LSTM mathematical model, which is able to take into account the temporal sequence of data, has proven to be effective in predicting trajectories, which can significantly reduce the risk of collision between a robot and a person and increase safety in a shared working environment. However, experiments have revealed that with fast and complex movements, the prediction accuracy decreases, which may require further optimization of the model or integration with other algorithms to improve performance under dynamic changes. The obtained results emphasize the need to adapt LSTM to specific usage scenarios, which opens up opportunities for further research in the direction of improving the accuracy and stability of forecasts. In general, the application of LSTM for this task can be an important step in the development of human-robot collaboration technologies, contributing to the improvement of efficiency and safety in production processes within the framework of the concept of Industry 5.0.

References:

1. Bortnikova, V., & et al. (2019). Structural parameters influence on a soft robotic manipulator finger bend angle simulation. In 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), IEEE, 35-38.
2. Samoilenko, H., & et al. (2024). Review for Collective Problem-Solving by a Group of Robots. Journal of Universal Science Research, 2(6), 7-16.
3. Gurin, D., & et al. (2024). Using the Kalman Filter to Represent Probabilistic Models for Determining the Location of a Person in Collaborative Robot Working Area. Multidisciplinary Journal of Science and Technology, 4(8), 66-75.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

4. Abu-Jassar, A., & et al. (2023). Obstacle Avoidance Sensors: A Brief Overview. Multidisciplinary Journal of Science and Technology, 3(5), 4-10.

5. Yevsieiev, V., & et al. (2024). The Canny Algorithm Implementation for Obtaining the Object Contour in a Mobile Robot's Workspace in Real Time. Journal of Universal Science Research, 2(3), 7–19.

6. Gurin, D., & et al. (2024). Effect of Frame Processing Frequency on Object Identification Using MobileNetV2 Neural Network for a Mobile Robot. Multidisciplinary Journal of Science and Technology, 4(8), 36-44.

7. Bortnikova, V., & et al. (2019). Mathematical model of equivalent stress value dependence from displacement of RF MEMS membrane. In 2019 IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), IEEE, 83-86.

8. Baker, J. H., Laariedh, F., Ahmad, M. A., Lyashenko, V., Sotnik, S., & Mustafa, S. K. (2021). Some interesting features of semantic model in Robotic Science. SSRG International Journal of Engineering Trends and Technology, 69(7), 38-44.

9. Sotnik, S., Mustafa, S. K., Ahmad, M. A., Lyashenko, V., & Zeleniy, O. (2020). Some features of route planning as the basis in a mobile robot. International Journal of Emerging Trends in Engineering Research, 8(5), 2074-2079.

10. Matarneh R., & et al. (2017). Speech Recognition Systems: A Comparative Review. Journal of Computer Engineering (IOSR-JCE), 19(5), 71–79.

11. Maksymova, S., Matarneh, R., Lyashenko, V. V., & Belova, N. V. (2017). Voice Control for an Industrial Robot as a Combination of Various Robotic Assembly Process Models. Journal of Computer and Communications, 5, 1-15.

12. Lyashenko, V., Abu-Jassar, A. T., Yevsieiev, V., & Maksymova, S. (2023). Automated Monitoring and Visualization System in Production. International Research Journal of Multidisciplinary Technovation, 5(6), 9-18.

13. Abu-Jassar, A. T., Attar, H., Lyashenko, V., Amer, A., Sotnik, S., & Solyman, A. (2023). Access control to robotic systems based on biometric: the generalized model and its practical implementation. International Journal of Intelligent Engineering and Systems, 16(5), 313-328.

14. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2023). Generalized Procedure for Determining the Collision-Free Trajectory for a Robotic Arm. Tikrit Journal of Engineering Sciences, 30(2), 142-151.

15. Ahmad, M. A., Sinelnikova, T., Lyashenko, V., & Mustafa, S. K. (2020). Features of the construction and control of the navigation system of a mobile robot. International Journal of Emerging Trends in Engineering Research, 8(4), 1445-1449.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

16. Lyashenko, V., Laariedh, F., Ayaz, A. M., & Sotnik, S. (2021). Recognition of Voice Commands Based on Neural Network. TEM Journal: Technology, Education, Management, Informatics, 10(2), 583-591.

17. Sotnik, S., & et al.. (2022). Agricultural Robotic Platforms. International Journal of Academic Engineering Research, 6(4), 14-21.

18. Lyashenko, V., & et al.. (2021). Semantic Model Workspace Industrial Robot. International Journal of Academic Engineering Research, 5(9), 40-48.

19. Sotnik, S., & et al.. (2022). Analysis of Existing Infliences in Formation of Mobile Robots Trajectory. International Journal of Academic Information Systems Research, 6(1), 13-20.

20. Sotnik, S., & et al.. (2022). Modern Industrial Robotics Industry. International Journal of Academic Engineering Research, 6(1),. 37-46.

21. Lyashenko, V., & et al.. (2021). Modern Walking Robots: A Brief Overview. International Journal of Recent Technology and Applied Science, 3(2), 32-39.

22. Yevsieiev, V., & et al. (2024). Building a traffic route taking into account obstacles based on the A-star algorithm using the python language. Technical Science Research In Uzbekistan, 2(3), 103-112.

23. Gurin, D., & et al. (2024). MobileNetv2 Neural Network Model for Human Recognition and Identification in the Working Area of a Collaborative Robot. Multidisciplinary Journal of Science and Technology, 4(8), 5-12.

24. Yevsieiev, V., & et al. (2024). Object Recognition and Tracking Method in the Mobile Robot's Workspace in Real Time. Technical science research in Uzbekistan, 2(2), 115-124.

25. Funkendorf, A., & et al. (2019). 79 Mathematical Model of Adapted Ultrasonic Bonding Process for MEMS Packaging. In 2019 IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), IEEE, 79-82.

26. Gurin, D., & et al. (2024). CAMShift Algorithm for Human Tracking in the Collaborative Robot Working Area. Journal of Universal Science Research, 2(8), 87–101.

27. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2021). Neural networks as a tool for pattern recognition of fasteners. International Journal of Engineering Trends and Technology, 69(10), 151-160.

28. Abu-Jassar, A. T., Al-Sharo, Y. M., Lyashenko, V., & Sotnik, S. (2021). Some Features of Classifiers Implementation for Object Recognition in Specialized

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

Computer systems. TEM Journal: Technology, Education, Management, Informatics, 10(4), 1645-1654.

29. Putyatin, Y. P., & et al.. (2016) The Pre-Processing of Images Technique for the Material Samples in the Study of Natural Polymer Composites. American Journal of Engineering Research, 5(8), 221-226.

30. Kobylin, O., & Lyashenko, V. (2014). Comparison of standard image edge detection techniques and of method based on wavelet transform. International Journal, 2(8), 572-580.

31. Lyashenko, V., Kobylin, O., & Ahmad, M. A. (2014). General methodology for implementation of image normalization procedure using its wavelet transform. International Journal of Science and Research (IJSR), 3(11), 2870-2877.

32. Mustafa, S. K., Yevsieiev, V., Nevliudov, I., & Lyashenko, V. (2022). HMI Development Automation with GUI Elements for Object-Oriented Programming Languages Implementation. SSRG International Journal of Engineering Trends and Technology, 70(1), 139-145.

33. Matarneh, R., Tvoroshenko, I., & Lyashenko, V. (2019). Improving Fuzzy Network Models For the Analysis of Dynamic Interacting Processes in the State Space. International Journal of Recent Technology and Engineering, 8(4), 1687-1693.

34. Tvoroshenko, I., Lyashenko, V., Ayaz, A. M., Mustafa, S. K., & Alharbi, A. R. (2020). Modification of models intensive development ontologies by fuzzy logic. International Journal of Emerging Trends in Engineering Research, 8(3), 939-944.

35. Lyashenko, V. V., Matarneh, R., & Deineko, Z. V. (2016). Using the Properties of Wavelet Coefficients of Time Series for Image Analysis and Processing. Journal of Computer Sciences and Applications, 4(2), 27-34.

36. Гиренко, А. В., Ляшенко, В. В., Машталир, В. П., & Путятин, Е. П (1996). Методы корреляционного обнаружения объектов. Харьков: АО “БизнесИнформ, 112.

37. Lyashenko, V., Matarneh, R., & Kobylin, O. (2016). Contrast modification as a tool to study the structure of blood components. Journal of Environmental Science, Computer Science and Engineering & Technology, 5(3), 150-160.

38. Lyashenko, V. V., Deineko, Z. V., & Ahmad, M. A. Properties of wavelet coefficients of self-similar time series. In other words, 9, 16.

39. Lyubchenko, V., & et al.. (2016). Digital image processing techniques for detection and diagnosis of fish diseases. International Journal of Advanced Research in Computer Science and Software Engineering, 6(7), 79-83.

ISSN (E): 2181-4570 ResearchBib Impact Factor: 6,4 / 2023 SJIF 2024 = 5.073/Volume-2, Issue-9

40. Lyashenko, V. V., Matarneh, R., Kobylin, O., & Putyatin, Y. P. (2016). Contour Detection and Allocation for Cytological Images Using Wavelet Analysis Methodology. International Journal, 4(1), 85-94.

41. Abu-Jassar, A. T., Attar, H., Amer, A., Lyashenko, V., Yevsieiev, V., & Solyman, A. (2024). Development and Investigation of Vision System for a Small-Sized Mobile Humanoid Robot in a Smart Environment. International Journal of Crowd Science.

42. Uchqun o'g'li, B. S., Valentin, L., & Vyacheslav, L. (2023). Preprocessing of digital images to improve the efficiency of liver fat analysis. Multidisciplinary Journal of Science and Technology, 3(1), 107-114.

43. Drugarin, C. V. A., Lyashenko, V. V., Mbunwe, M. J., & Ahmad, M. A. (2018). Pre-processing of Images as a Source of Additional Information for Image of the Natural Polymer Composites. Analele Universitatii'Eftimie Murgu', 25(2).

44. Gualtieri, L., & et al. (2021). Emerging research fields in safety and ergonomics in industrial collaborative robotics: A systematic literature review. Robotics and Computer-Integrated Manufacturing, 67, 101998.

45. Vicentini, F. (2021). Collaborative robotics: a survey. Journal of Mechanical Design, 143(4), 040802.

46. Sherwani, F., & et al. (2020). Collaborative robots and industrial revolution 4.0 (ir 4.0). In 2020 International Conference on Emerging Trends in Smart Technologies (ICETST), IEEE, 1-5.

47. Liu, L., & et al. (2024). Application, development and future opportunities of collaborative robots (cobots) in manufacturing: A literature review. International Journal of Human–Computer Interaction, 40(4), 915-932.

48. Franklin, C. S., & et al. (2020). Collaborative robotics: New era of human–robot cooperation in the workplace. Journal of Safety Research, 74, 153-160.

49. Knudsen, M., & Kaivo-Oja, J. (2020). Collaborative robots: Frontiers of current literature. Journal of Intelligent Systems: Theory and Applications, 3(2), 13-20.

50. Bi, Z. M., & et al. (2021). Safety assurance mechanisms of collaborative robotic systems in manufacturing. Robotics and Computer-Integrated Manufacturing, 67, 102022.

51. Chemweno, P., & et al. (2020). Orienting safety assurance with outcomes of hazard analysis and risk assessment: A review of the ISO 15066 standard for collaborative robot systems. Safety Science, 129, 104832.