Modeling of a Neural Network-Based Motor Position Controller in a System for Tracking Objects of Complex Shapes

Oleksandr Laktionov[1,*,†], Alina Yanko[2, †], Alina Hlushko[3,†] and Victor Krasnobayev[4,†]

[1] National University «Yuri Kondratyuk Poltava Polytechnic», Pershotravnevyj Ave 24, Poltava, 36011, Ukraine

Abstract

This research is dedicated to enhancing the efficiency of a system for tracking objects of complex shapes through the integration of movable cameras and a neural network-based motor position controller. The aim of this work is to ensure accurate and reliable real-time object tracking. In this study, a system for tracking objects of complex shapes was developed and investigated, utilizing a camera mounted on an electric motor, with and without neural network-based motor position controller. A key aspect of the research is the training of a neural network model based on electric motor position data during tracking. The model's output data are used to predict the electric motor's position, enabling proactive motion correction and improved tracking accuracy. A distinctive feature of this research is the adaptation of the neural network-based motor position controller for localized use in a system for tracking objects of complex shapes, specifically designed to address current challenges faced by regional industrial enterprises. The practical value of this work lies in the potential application of the developed system in industry and educational processes to enhance technical safety. The system's flexibility allows for its use with or without a neural network-based motor position controller, ensuring rapid configuration and adaptation to various conditions. The current prototype utilizes a 2MP camera, and while the integration of an LSTM-based motor position controller showed a minor reduction in the standard deviation of positioning errors (from 168.88 to 164.11), future work will focus on incorporating higher-resolution cameras with improved low-light performance and further optimization of the neural network architecture and training dataset to enhance tracking accuracy.

Keywords

neural network-based controller, artificial intelligence, computer vision, object tracking, electric motor position prediction, neural network

1. Introduction

In modern automated systems and robotics, object tracking plays a pivotal role, finding applications in various domains ranging from video surveillance to automated production control. Precise tracking, especially of objects with complex shapes, necessitates continuous and real-time correction of movable mechanism positions. Electric motors are a crucial component of such systems, providing high-precision positioning, yet their stable operation requires the use of sophisticated control algorithms capable of mitigating diverse external influences and errors [1].

Current approaches to electric motor control include the use of traditional controllers, but to ensure high accuracy and adaptability in unpredictable environmental changes and object variations, more advanced methods such as neural network-based controllers are essential. Neural

ICST-2025: Information Control Systems & Technologies, September 24-26, 2025, Odesa, Ukraine

* Corresponding author.
† These authors contributed equally.
itm.olaktionov@nupp.edu.ua (O. Laktionov); al9_yanko@ukr.net (A. Yanko); glushk.alina@gmail.com (A. Hlushko); v.a.krasnobaev@gmail.com (V. Krasnobayev)
0000-0002-5230-524X (O. Laktionov); 0000-0003-2876-9316 (A. Yanko); 0000-0002-4086-1513 (A. Hlushko); 0000-0001-5192-9918 (V. Krasnobayev)

Network-Based Controllers, built upon neural networks, demonstrate the ability to adapt to changing conditions, optimizing control parameters in real time. They enable the reduction of noise, positional estimation errors, and other unforeseen factors that arise during the tracking of objects with complex geometries. This minimizes static and dynamic positioning errors, enhances system resistance to external influences, and ensures optimal real-time operation.

The task of tracking complex-shaped objects in the context of regional industrial enterprises is particularly relevant, where the accuracy and reliability of video surveillance systems are critical for ensuring the safety and efficiency of production processes. In this context, the development of a tracking system utilizing a Neural Network-Based Controller capable of predicting object motion and proactively adjusting the camera position is not only a scientific but also a practical necessity.

The foundation of this research is the development concept that combines computer vision with an actuator that adjusts the camera position based on object movement, integrated with a neural network model. The study is based on the principles of stacking, adaptive learning, and neural network control, allowing the integration of computer vision capabilities with precise actuator control. The research emphasizes the creation of a model for predicting signals several seconds ahead, enabling preemptive activation of the stepper motor. The scientific novelty of this work lies in adapting a well-established scientific approach to localize the use of neural network control in a tracking system to address specific challenges faced by regional industrial enterprises.

The aim of this research is to enhance the operational efficiency of computer vision models by implementing movable cameras and a tracking system with neural network control, thereby ensuring more accurate and reliable real-time tracking of complex-shaped objects.

To achieve this goal, the following research tasks were defined:

1. Develop a tracking system with and without a neural network-based controller, capable of independent operation.
2. Conduct an experimental study for comparative analysis of the effectiveness of both systems.

2. Analysis of existing scientific approaches

Early prototypes of tracking systems were implemented on Arduino controllers [2]. The software involved object detection followed by tracking. This technology enhances tracking accuracy through the use of cascaded classifiers [3]. Over time, the hardware and software have actively evolved and transformed into an integrated ecosystem incorporating artificial intelligence.

Artificial intelligence tools continuously learn from specific data, thereby updating the model [4]. Therefore, contemporary research is oriented towards developing new artificial intelligence models to address specific tasks. The process of model parameter identification is frequently considered [5]. This allows for the discovery of new parameters and enhances prediction accuracy.

Prediction accuracy is also improved by creating neural network controllers, as exemplified in [6]. A notable feature of the proposed solution is the existence of a real-world model from which data is collected and fed into the artificial intelligence model. This work is among the pioneering efforts that have unlocked new possibilities for artificial intelligence applications. In addition to developing models that predict the position of actuators, the selection and integration of other equipment, particularly video cameras, with the control system is crucial. Study [7] proposes a PTZ camera control system that automatically detects and tracks moving objects in real-time, utilizing their center, direction of motion, distance, and speed, regardless of the camera's focusing function. Implemented on a TI DM6446 DSP processor, this system demonstrates high efficiency in tracking high-speed vehicles. The study also highlights the limitations of software camera focus, underscoring the necessity for a motor-driven movement system.

Article [8] focuses on the development of Complementary metal-oxide semiconductor image sensor and its applications in aerospace, medical and automotive fields. The sensor can be created in specialized software and manufactured at the enterprise. Such sensors can expand the capabilities of computer vision systems in interaction with other equipment, primarily cameras. Therefore, this study expands on previous work [7].

For construction applications, [9] proposes an automated tracking system for construction machinery on unmanned construction sites, combining image processing and machine learning techniques to improve accuracy and reliability. This system utilizes a platform that adjusts direction as needed, but via manual command. Although the algorithm provides stable and continuous imaging, it is hampered by the issue of manual control. Research [10] introduces a novel Position Alignment Method (PAM) that automatically, accurately, and rapidly aligns coordinate systems, ensuring error-free calibration in remote camera control. Experimental comparisons show that PAM outperforms manual methods in terms of accuracy, stability, and operational speed, and is more flexible for use in telerobotic camera control.

The use of motors increases the camera's range of motion, but in [11], an algorithm for automatic detection, tracking, and zooming of active targets using a camera with an already wide range of motion is presented, improving the resolution of distant objects. The proposed system optimizes disk space usage by stopping recording when no targets are present and provides adaptive tracking of multiple objects with motion prediction to minimize image quality loss and reduce the need for camera movement. Study [12] demonstrates a developed automatic position correction module for an image inspection system, which enables the camera to adjust its pose and position based on detected object displacement or rotation errors. Results show that the system with position correction significantly enhances productivity by automating the optical quality inspection process.

Any platform movement destabilizes the camera, reducing image quality and tracking accuracy. The visual tracking system for mobile robots proposed in [13] stabilizes images during motion using a combination of feedforward control from gyroscope and encoder data (VOR) and periodic feedback correction (OKR). Study [14] presents a visual tracking system for a mobile robot that uses stereo camera and motion sensor data to maintain a line of sight to a stationary target. Vision-based compensation is applied to correct motion measurement errors, activated when the robot stops or moves slowly, ensuring high tracking accuracy without overloading the system. The background suppression algorithm, considering camera motion to minimize the impact of its oscillations caused by wind or heavy transport vibrations, which is especially critical at high focal lengths, is presented in [15]. During motor movement, the proposed data processing-based stabilization approach to compensate for camera rotation in real-time in [16] improves tracking accuracy of features and estimation of independent camera motion. Experiments show that stabilization increases accuracy by 27.37% for feature tracking and 34.82% for independent motion estimation, and reduces processing time by 25%.

The use of motors is also necessary for calibrating installed cameras. Study [17] proposes a rotation-based camera and gyroscope calibration method that eliminates the need for targets and accurately estimates intrinsic camera parameters and extrinsic system parameters. The method is verified on real data from a low-cost platform, making it suitable for lightweight robotic platforms equipped with cameras and gyroscopes.

The developed camera stabilization control system on a gimbal for unmanned aerial vehicles (UAVs), used for tasks such as target tracking, surveillance, and aerial photography in [18], shows that traditional PID control is less effective compared to PID control with settings tuned by the PSO algorithm.

In the context of enhancing the reliability and efficiency of data processing systems, it is crucial to use methods that ensure error resistance and high performance [19]. In this context, Residue number systems (RNS) can play a key role. Prior studies [20] and [21] analyze the impact of Residue Number Systems on error resistance and the efficiency of computer systems, particularly in the context of error diagnostics in data processing devices.

Considering the advantages of RNS in providing parallel computations and error resistance, their application in tracking systems can enhance data processing speed and reliability, especially in industrial settings where speed and accuracy are critical. Further research will focus on integrating RNS into tracking systems to improve their performance.

3. Tools for development of a neural network-based motor position controller in a system for tracking objects of complex shapes

This research was conducted using a Raspberry Pi single-board computer, a Nema 17 stepper motor, a TB6600 stepper motor driver, and an HD 2MP video camera, as illustrated in Fig. 1.

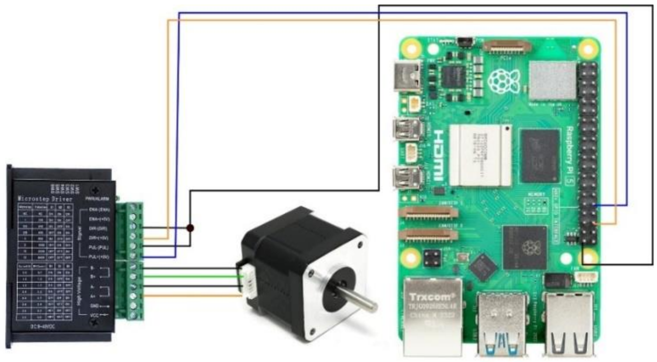

Figure 1: Prototype of the investigated electrical circuit equipment based on the Raspberry Pi single-board computer.

Two approaches to action strategies were considered. The first action strategy involved developing an object tracking system with camera movement when the tracked object reached the edge of the graphical interface. The second action strategy involved developing an object tracking system with camera movement controlled by a neural network when the tracked object reached the edge of the graphical interface.

As neural networks, Facebook Prophet [22] and Long Short-Term Memory (LSTM) [23] were studied. To determine the signal shape and train the neural network, a single video sequence was used, and the tracking object's movement data were recorded in a .csv file for in-depth analysis. The input data included the object's displacement from the center of the graphical interface and time in seconds, which were recorded in a Table 1.

Table 1
Comparative analysis of the displacement of the investigated area's position and stepper motor operating time

No.	Without neural network-based controller		With neural network-based controller	
	Displacement of the investigated area's position	Time, seconds	Displacement of the investigated area's position	Time, seconds
1	-70	0.10867	-70	0.10867
300	120	83.682	120	83.553
	Standard deviation 168.88		Standard deviation 164.11	

Justification of the neural network model selection required the use of a test video sequence (self-created) to capture the signal shape, as shown in Fig. 2.

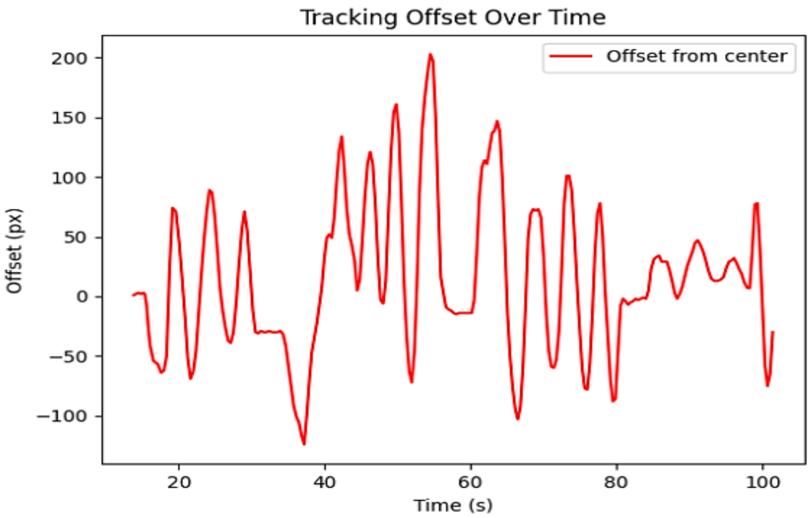

Figure 2: Signal shape of object tracking during object movement along the x-axis left and right for neural network selection.

In the research process, the Python programming language [24] and external libraries installed in the single-board computer's environment, myenv, were used, as depicted in Fig. 3.

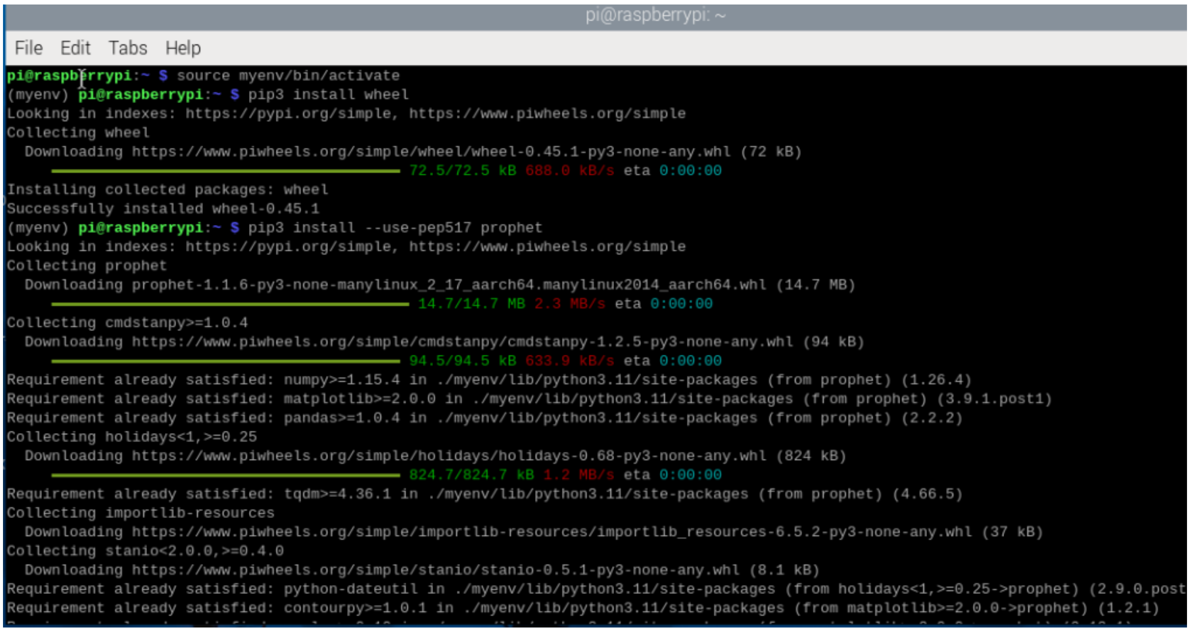

Figure 3: Setting up the Prophet neural network library.

The primary video stream processing library was cv [25]. The gpiozero library [26] was used for stepper motor control. The analysis of accumulated data involved several standard libraries, including pandas, numpy, and matplotlib, which are implemented in the programming language [24]. Neural networks required the use of the sklearn [27] and tensorflow [28] libraries. In addition to these library packages, the Prophet library [22] was used. To build the models, the methodology was used [27], [28].

The configuration of the Prophet library, with the prior installation of additional libraries necessary for Prophet to function, particularly plotly [29] for graphical interpretation of the results, is shown in Fig. 3.

The developed models were saved in a .h5 file for use on the Raspberry Pi. Thus, the basic tools for conducting the research were prepared.

The main idea of tracking is to maintain the tracked object, especially during camera movement by the stepper motor. For this purpose, an interface with specific functionality was created. The buttons included settings for region of interest (ROI): w for forward, s for backward, a for left, d for right, enter to start tracking, +/- for scaling, q to end tracking, r to start recording offset and time, and t to finish recording. Key elements of the tracking system are the tracking algorithms, which in this study included Channel and Spatial Reliability Tracker, Kernelized Correlation Filters, Minimum Output Sum of Squared Error Filter, and Multiple Instance Learning.

The development of the tracking or object following system was carried out according to the following subtasks:

1. The ROI should appear in the center of the graphical interface, and upon ending tracking (button q), the ROI should return to the center position.
2. The operator selects the tracking area and adjusts the ROI scale.
3. Upon starting tracking, the program should determine the distance from the ROI to the left and right boundaries and to the center of the graphical interface.
4. If the distance between the ROI and the boundary is less than 20 px, the camera should move 1 step left or right (depending on the ROI position).
5. When the distance from the ROI to the edge of the interface is too large, the stepper motor should perform one step at a time to avoid losing the detected object.
6. Do not accelerate the stepper motor's movement, even when the ROI is close to the edge of the graphical interface. Limit the number of steps.
7. Display information about the ongoing action on the interface.
8. Use the gpiozero library to control the stepper motor. Stepper motor configuration: dirPin = 16, stepPin = 12, MAX_ANGLE = 30 # -30 to +30 degrees, STEP_ANGLE = 1.8 # Stepper motor step in degrees (e.g., 200 steps per revolution -> 1.8 degrees per step).
9. To determine the stepper motor's rotation direction, self.direction.value = 0, use the condition if direction > 0 else 1 (1 for clockwise rotation, otherwise counterclockwise).

The software also had requirements for debugging the implemented program texts. During debugging data recording, if the object is lost, recording should stop, and resume when the object reappears in the frame. If the object is lost and cannot be found, the operator should exit tracking mode, reconfigure tracking mode, and continue data recording.

Comparative analysis of the studied values was carried out using the standard deviation criterion (see Table 1).

4. Results of modeling a neural network-based motor position controller in a system for tracking objects of complex shapes

4.1. Development and debugging of the basic functionality of object tracking without a neural network controller

According to the research program, the primary step involved the implementation of software for object tracking with a video camera displacement system upon the tracked object reaching the extreme position of the graphical interface. The software implementation was carried out in several files, specifically import_cv2.py, import_cv2-1.py, import_cv2-2.py, ..., import_cv2-4-1.py.

During the debugging of the proposed solution, implemented in the import_cv2.py file, certain errors arose, notably the selection of an excessive number of stepper motor steps. This led to a technical loss of the detection object, as the camera rotation angle was too large. Introducing the camera's limitation during the discussion of the initial hardware setup and early challenges makes sense. It provides context for potential issues encountered during the debugging phase, even if those specific issues weren't directly caused by the camera resolution. It is important to note that the system at this stage utilized a 2MP camera, the resolution of which, while sufficient for initial testing, presented an inherent limitation in capturing fine details and could potentially impact tracking accuracy, especially for distant or small objects.

The subsequent version of the software implementation, import_cv2-1.py, addressed the aforementioned issues. For instance, a decision was made to create a bounding box, 10% smaller on the left and right sides than the main graphical interface. Upon approaching the region of interest to the frame, the stepper motor with the camera was to be activated and adjust the camera position. However, this idea was also imperfect, as upon the region of interest approaching the frame, if the tracking object moved beyond the video stream, it was lost.

In the import_cv2-2.py version, the number of stepper motor steps for camera movement per unit time was reduced. To enhance control over camera movements, the detection threshold of the bounding box of the studied area was increased to 15%. Text messages regarding the stepper motor speed and the tracking object position were added to the interface. Consequently, the following limitations were observed: if the detection object moves along the X-axis and exits the study area, the stepper motor does not rotate the camera. If the detection object is in the center of the study area, the motor rotates the camera by a specified step. These and other contradictions were addressed in subsequent software versions, with the desired result achieved only in import_cv2-4-1.py, as shown in Fig. 4.

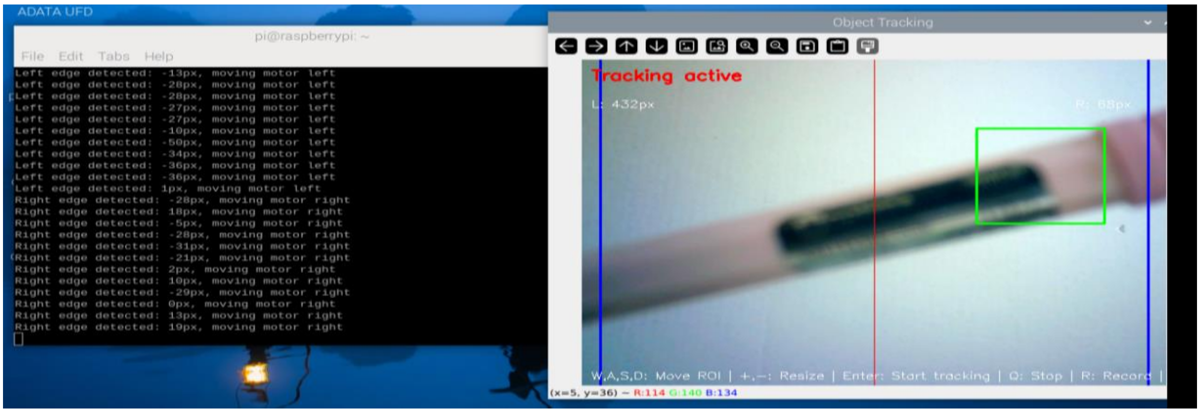

Figure 4: Results of the proposed solution in laboratory conditions, command line (left), graphical interface of the tracking system (right).

As shown in Fig. 4, the graphical interface has a classic layout. Control commands are located in the lower part, and status messages are displayed in the upper part. The tracking object is in the center of the frame.

4.2. Rationale for selecting models of a neural network controller for tracking objects of complex shapes and their construction

The subsequent part of the research was dedicated to the development and selection of an optimal neural network controller model, where two models, Prophet and LSTM, were compared. To verify the functionality of the Prophet model on a single-board computer, the first simple program was implemented (Fig. 5).

As shown in Fig. 5, the model functions. At the next stage of the research, according to the research program, data accumulation was performed. As can be seen from the graph, the signal is close to sinusoidal, so a sinusoidal signal form was generated for 300 seconds (Fig. 6). The main goal of creating the model is to predict the signal a few seconds ahead, so that the stepper motor is activated in advance.

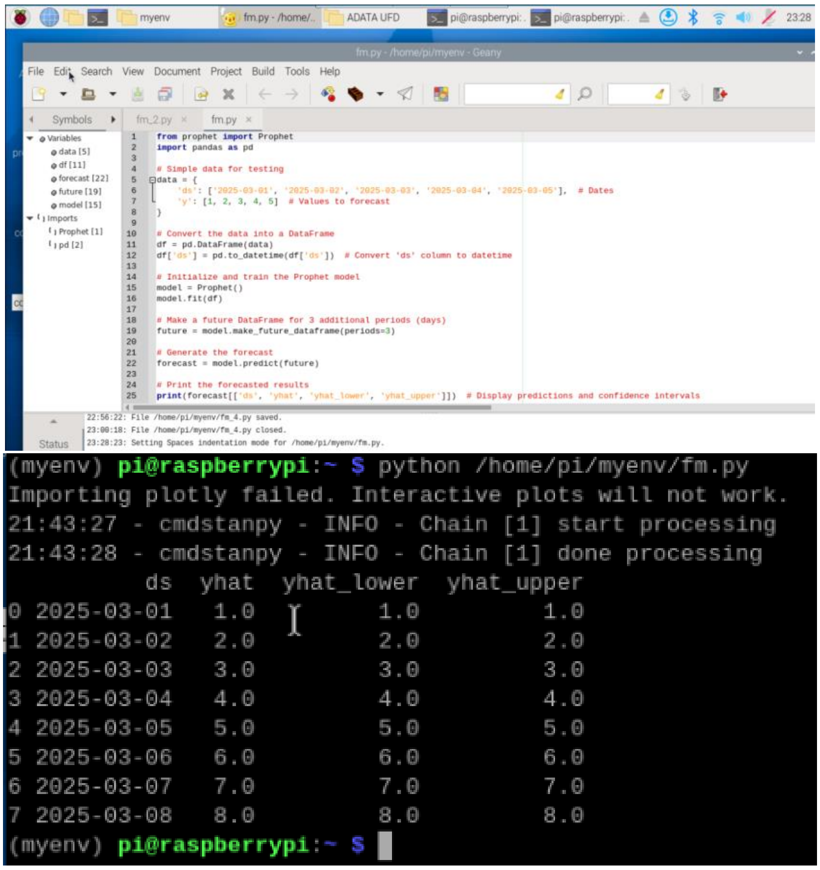

Figure 5: Testing the functionality of the Prophet model on a Raspberry Pi single-board computer.

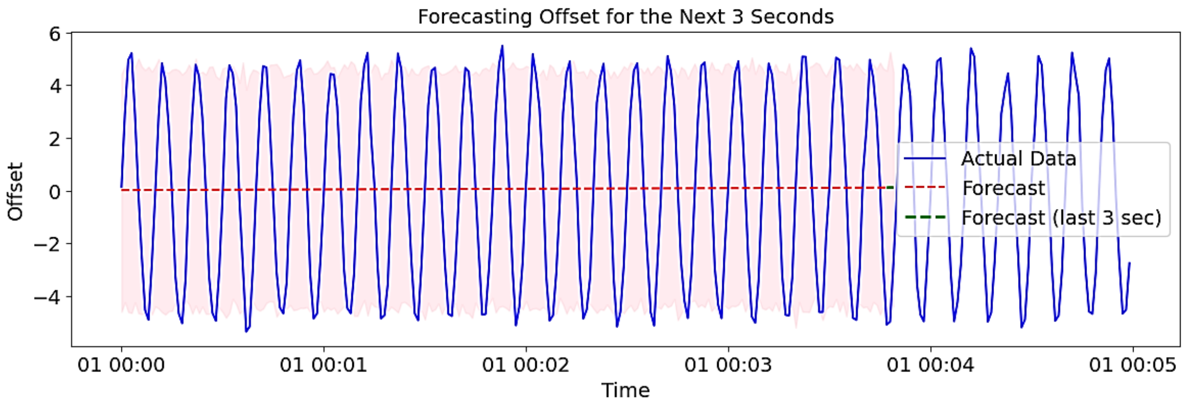

Figure 6: Rationale for selecting the Prophet artificial intelligence model based on a theoretical dataset.

Despite various ways of using the Prophet model, it does not reproduce the input signal in the form of a sinusoid, so it will not work adequately in the system being developed. Attempts to represent the new record Facebook Prophet=g(t·x1)·s(t·x2) ·h(t·x3)·noise as a mathematical notation and implement it programmatically did not show the desired result. Additionally, Auto Regressive Integrated Moving Average tools were used [30], but it has limitations on the number of variables. Further, the LSTM model with the Adam optimizer was used, training was performed on 50 epochs, with batch_size=16. Before training the network, the classic steps of its construction were performed [31]. The sample was differentiated into training/test in a ratio of 75/25. The criteria for the quality of the model construction were the coefficients of determination on both subsamples and the value of loss='mse'. The actual and predicted values using LSTM model as shown in Fig. 7.

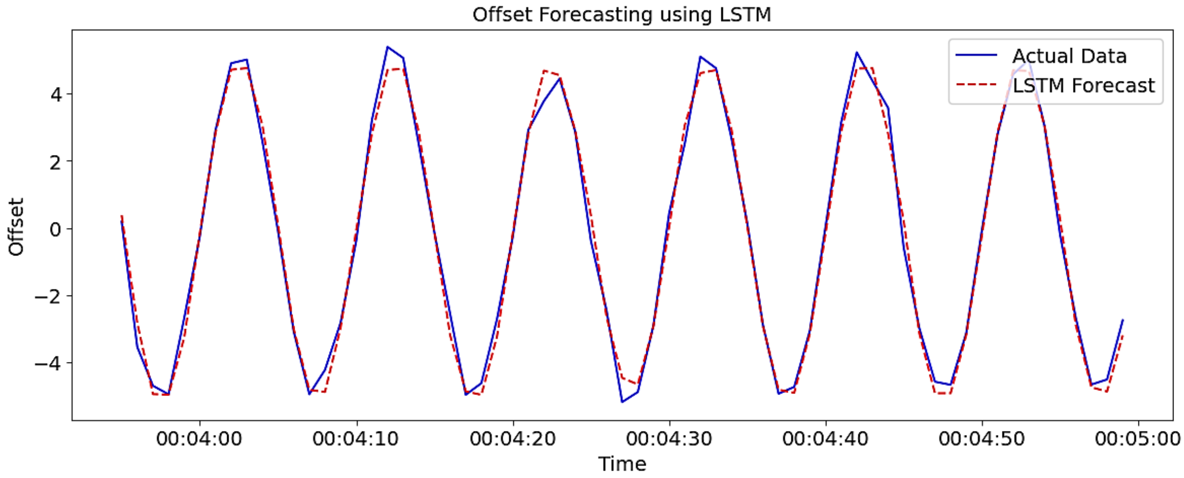

Figure 7: Rationale for selecting the LSTM artificial intelligence model based on a theoretical dataset.

As can be seen from Fig. 7, the actual and predicted values almost coincide, as evidenced by the calculated data. The coefficients of determination on the training/test samples are 0.98/0.98, which indicates the absence of overfitting, with loss: 0.0038 - val_loss: 0.0054.

5. Experimental research and practical application

Following the theoretical modeling, experimental modeling of the tracking system was conducted. For this purpose, a dataset was accumulated, as shown in Fig. 8.

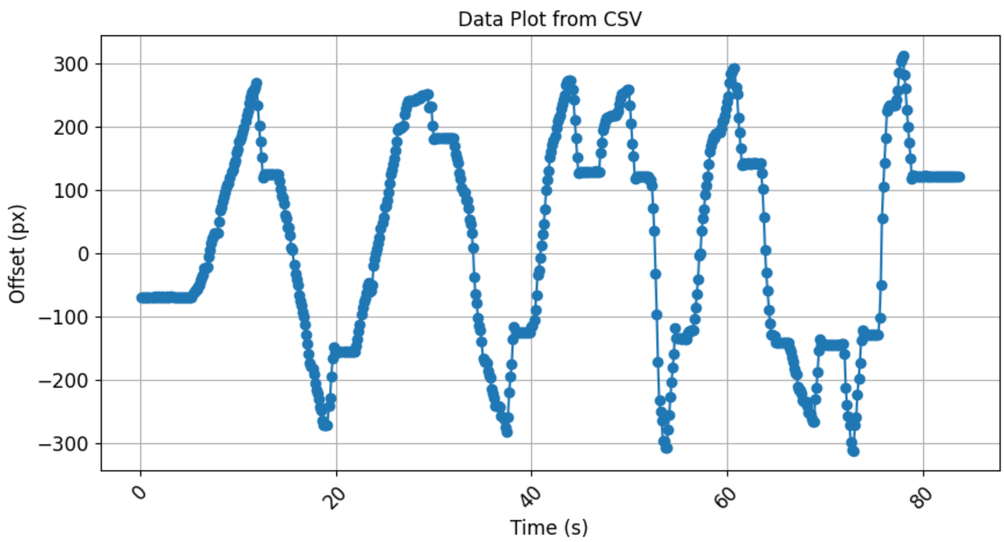

Figure 8: Example of experimentally accumulated data submitted to LSTM model for training.

The data in Fig. 8 were initially examined for gaps, analyzed, and fed into the neural network, as shown in Fig. 9.

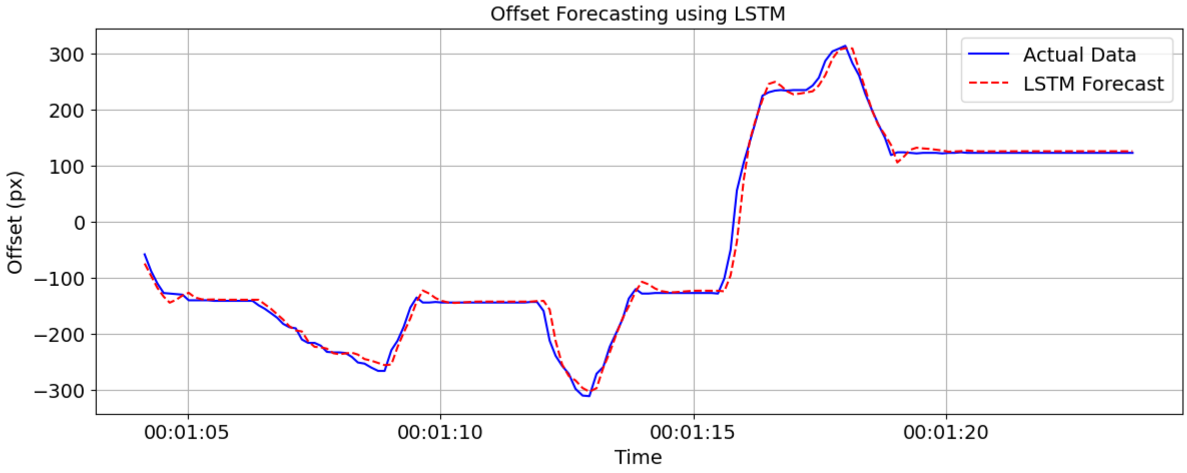

Figure 9: Example of prediction using LSTM.

The constructed model does not exhibit signs of overfitting, as evidenced by R2 values of 0.99 for both subsamples. The graphical interpretation of the stepper motor displacement prediction result indicates high accuracy, as the actual and predicted data coincide. This is also confirmed by the training error loss: 0.0011 - val_loss: 0.0023.

Based on the comparative analysis using the standard deviation criterion, the tracking system without a neural network controller demonstrates a standard deviation of 168.88, while the system with a neural network controller shows a standard deviation of 164.11. This allows for predicting the motor activation time for camera displacement depending on the position of the studied detection object. Let us apply the created solutions to practical tasks. The tracking distance was investigated from 0 to 300 meters. Algorithms such as Channel and Spatial Reliability Tracker, Kernelized Correlation Filters [32] and others do not detect images at a distance of 200-300 meters with a region of interest size of 30x30 px, even with image zooming. However, various factors influence this, including the object size. The quality of daytime object tracking correlates with natural factors (sunlight entering the camera lens), the object rapidly changing its trajectory, and tracking losing the object. As practice shows, the proposed solution works at a distance of up to 50-60 meters in daylight. For example, a car of any color is tracked in daylight, as shown in Fig. 10.

Figure 10: Example of testing the operating range of the tracking system (distance 50-60 meters).

As shown in Fig. 10, the system tracks the car and person even in the presence of obstacles, such as trees. The solution can be used in industry or education for safety support.

6. Analysis of limitations and shortcomings of the system

The developed system for tracking objects of complex shapes, while demonstrating high efficiency under certain conditions, has a number of limitations that need to be considered for its further improvement and practical application.

At this stage, the development does not include a specialized case for transportation, which limits its mobility and usability in field conditions. To expand the scope of application of the system, it is necessary to develop a reliable and convenient case that will ensure the protection of components during transportation and rapid deployment on-site [33]. To assess the economic feasibility and reliability of the system, it is necessary to conduct a detailed analysis of the cost of its components (camera, single-board computer, electric motor) and study their resistance to external influences [34]. Differentiation of components will allow determining the optimal ratio between cost and quality. The effectiveness of the system can vary significantly depending on the lighting level and the type of objects being tracked. To ensure stable operation of the system in different conditions, it is necessary to conduct experiments with different lighting levels (day, night, artificial) and different detection objects (people, vehicles, industrial parts).

At this stage, the system is controlled using a keyboard, which limits its convenience and the possibility of remote control. To expand the functionality of the system, it is necessary to implement remote control using radio signals, Wi-Fi, or other wireless technologies, such as Mesh Networking [35]. To ensure autonomous operation of the system in field conditions, it is necessary to use specialized power modules, such as batteries or solar panels. The choice of power module should take into account the power consumption of the system components and the duration of autonomous operation. The effectiveness of using a neural network controller depends on the quality and volume of training data. To ensure adequate functioning of the system in different scenarios, it is necessary to pre-train the neural network on a large dataset that reflects different types of movements and observation conditions.

Considering these limitations and shortcomings, further research will focus on their elimination and expansion of the functionality of the developed system.

7. Conclusion

The findings of this research demonstrate the successful development of a system for tracking complex-shaped objects within a 50-60 meter range under daylight conditions. The system effectively employs a 2MP camera and a stepper motor for automated camera position correction based on object movement at the interface edges. The implementation allows for operation both with and without an LSTM-based neural network controller, enabling a comparative analysis of their effectiveness. A comparative analysis of the tracking system with and without the LSTM-based neural network revealed a marginal reduction in the standard deviation (164.11 vs. 168.88), suggesting a potential for enhanced positioning accuracy; however, it necessitates further rigorous optimization of tracking parameters, filtering, and signal smoothing. Optimization of operation is also possible by using more powerful equipment, such as video cards or acceleration.

Future work will focus on a detailed experimental setup refinement, expanding system functionality through remote control and autonomous power, developing a protective transportation case, and a thorough investigation into cost-effectiveness and reliability across diverse lighting conditions and object types. Critically, upcoming research will prioritize enhancing the neural network's performance through experimentation with various architectures (e.g., CNN-LSTM hybrids), increasing the size and diversity of the training dataset, and applying advanced optimization techniques to improve the motor position prediction accuracy.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] B. Boriak, A. Yanko, O. Laktionov, Model of an automated control system for the positioning of radio signal transmission/reception devices, Radioelectronic and Computer Systems 4(112) (2024) 156–167. doi:10.32620/reks.2024.4.13.

[2] Face-Detection. URL: https://github.com/rizkydermawan1992/face-detection.
[3] O. Laktionov, A. Yanko, N. Pedchenko, Identification of air targets using a hybrid clustering algorithm, Eastern-European Journal of Enterprise Technologies 5(4(131)) (2024) 89–95. doi:10.15587/1729-4061.2024.314289.
[4] D. O'Shaughnessy, Understanding Automatic Speech Recognition, Computer Speech & Language (2023) 101538. doi:10.1016/j.csl.2023.101538.
[5] V. B. Mokin, A. V. Losenko, A. R. Yascholt, Information technology analysis and predicting a multiwave number of new COVID-19 disease based on prophet model, Visnyk of Vinnytsia Politechnical Institute 153(6) (2020) 65–75. doi:10.31649/1997-9266-2020-153-6-65–75.
[6] S. Vladov, A. Sachenko, V. Sokurenko, O. Muzychuk, V .Vysotska, Helicopters turboshaft engines neural network modeling under sensor failure, Journal of Sensor and Actuator Networks 13(5) (2024) 66. doi:10.3390/jsan13050066.
[7] Jeong, Cheol-Jun, Park, Goo-Man, Real-time auto tracking system using PTZ camera with DSP, International journal of advanced smart convergence 2(1) (2013) 32–35. doi:10.7236/IJASC.2013.2.1.032.
[8] G. Li, F. Zhao, Z. Wang, Z. Chen, The development and application of CMOS image sensor, Applied and Computational Engineering 7 (2023) 767–777. doi:10.54254/2755-2721/7/20230460.
[9] M. Fujitake, M. Inoue, and T. Yoshimi, Development of an Automatic Tracking Camera System Integrating Image Processing and Machine Learning, Journal of Robotics and Mechatronics 33(6) (2021) 1303–1314. doi:10.20965/jrm.2021.p1303.
[10] J. He, J. Huang, W. Zhang, F. Xu, Q. Liu, H. Li, PAM: Research on posture alignment method for camera robot system, International Journal of Advanced Robotic Systems 21(5) 2024. doi:10.1177/17298806241278908.
[11] S.-C. Hsia, S.-H. Wang, C.-M. Wei, C.-Y. Chang, Intelligent object tracking with an automatic image zoom algorithm for a camera sensing surveillance system, Sensors 22(22) (2022) 8791. doi:10.3390/s22228791.
[12] A. K. Bedaka, S.-C. Lee, A. M. Mahmoud, Y.-S. Cheng, C.-Y. Lin, A camera-based position correction system for autonomous production line inspection, Sensors 21(12) (2021) 4071. doi:10.3390/s21124071.
[13] J. Park, W. Hwang, W. Bahn, C. Lee, T. Kim, M. M. Shaikh, K. Kim, D. Cho, Pan/Tilt camera control for vision tracking system based on the robot motion and vision information, in: Proceedings of the 18th World Congress The International Federation of Automatic Control, volume 44 of IFAC, Milano, Italy, 2011, pp. 3165–3170. doi:10.3182/20110828-6-IT-1002.01781.
[14] W. Bahn, J. Park, C. Lee, T. Kim, T. Lee, M.M. Shaikh, K. Kim, D. Cho, A motion-information-based vision-tracking system with a stereo camera on mobile robots, in: Proceedings of the 5th International Conference on Robotics, Automation and Mechatronics (RAM), IEEE, Qingdao, China, 2011, pp. 252–257. doi:10.1109/RAMECH.2011.6070491.
[15] P. Mazurek, K. Okarma, Background suppression for video vehicle tracking systems with moving cameras using camera motion estimation, in: J. Mikulski, (Eds.), Proceedings of the 12th International Conference on Transport Systems Telematics, volume 329 of TST 2012, Springer, Berlin, Heidelberg, 2012, pp. 372–379. doi:10.1007/978-3-642-34050-5_42.
[16] J. P. Rodríguez-Gómez, J. R. M. -d. Dios, A. Ollero, G. Gallego, On the benefits of visual stabilization for frame- and event-based perception, IEEE Robotics and Automation Letters 9(10) (2024) 8802–8809. doi:10.1109/LRA.2024.3450290.
[17] P. Li, J. H. Ren, An Efficient Gyro-Aided Optical Flow Estimation in Fast Rotations With Auto-Calibration, IEEE Sensors Journal 18(8) (2018) 3391–3399. doi:10.1109/JSEN.2018.2810060.
[18] R. J. Rajesh, C. M. Ananda, PSO tuned PID controller for controlling camera position in UAV using 2-axis gimbal, in: Proceedings of the 2015 International Conference on Power and Advanced Control Engineering (ICPACE), Bengaluru, India, 2015, pp. 128–133. doi:10.1109/ICPACE.2015.7274930.
[19] S. Onyshchenko, A. Yanko, A. Hlushko, O. Maslii, A. Cherviak, Cybersecurity and improvement of the information security system, Journal of the Balkan Tribological

Association 29(5) (2023) 818–835. URL: https://scibulcom.net/en/article/L8nV7It2dVTBPX09mzWB.

[20] S. Onyshchenko, A. Yanko, A. Hlushko, Improving the efficiency of diagnosing errors in computer devices for processing economic data functioning in the class of residuals, Eastern-European Journal of Enterprise Technologies 5(4(125)) (2023) 63–73. doi:10.15587/1729-4061.2023.289185.

[21] A. Yanko, V. Krasnobayev, A. Martynenko, Influence of the number system in residual classes on the fault tolerance of the computer system, Radioelectronic and Computer Systems, 3(107) (2023) 159–172. doi:10.32620/reks.2023.3.13.

[22] Facebook/Prophet. URL: https://github.com/facebook/prophet.

[23] R. C. Staudemeyer, E. R. Morris, Understanding LSTM – a tutorial into long short-term memory recurrent neural networks, 2019. URL: https://arxiv.org/pdf/1909.09586.

[24] Python, 2025. URL: https://www.python.org/.

[25] OpenCV, 2025. URL: https://opencv.org/.

[26] Gpiozero, 2025. URL: https://gpiozero.readthedocs.io/en/latest/.

[27] Scikit-learn. Machine Learning in Python, 2025. URL: https://scikit-learn.org/.

[28] TensorFlow, 2025. URL: https://www.tensorflow.org/.

[29] Plotly, 2025. URL: https://plotly.com/.

[30] G. Sunitha, A. Sriharsha, Y. Olimjon, I. Mamatov, Interactive Visualization With Plotly Express, in: M. G. Galety, et al. (Eds.), Advanced Applications of Python Data Structures and Algorithms, IGI Global, 2023, pp. 182–206. doi:10.4018/978-1-6684-7100-5.ch009.

[31] Sh. Shalev-Shwartz, Sh. Ben-David (Ed.), Understanding machine learning: from theory to algorithms, Cambridge University Press, 2014. URL: https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/.

[32] A. H. Mbouombouo Mboungam, Y. Zhi, C. K. Fonzeu Monguen, Real-time tracking target system based on kernelized correlation filter in complicated areas, Sensors 24 (2024) 6600. doi:10.3390/s24206600.

[33] A. Yanko, N. Pedchenko, O. Kruk, Enhancing the protection of automated ground robotic platforms in the conditions of radio electronic warfare, Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu 6 (2024) 136–142. doi:10.33271/nvngu/2024-6/136.

[34] S. Onyshchenko, V. Skryl, A. Hlushko, O. Maslii, Inclusive Development Index. In: Onyshchenko, V., Mammadova, G., Sivitska, S., Gasimov, A. (eds), Proceedings of the 4th International Conference on Building Innovations (ICBI 2022), volume 299 of Lecture Notes in Civil Engineering, Springer, Cham, 2023, pp. 779–790. doi:10.1007/978-3-031-17385-1_66.

[35] P. Cardullo, D. J. Roio, Mesh Networking, OSF Preprints tepgc, Center for Open Science, 2019. doi:10.31219/osf.io/tepgc.