Acumen: International Journal of Multidisciplinary Research
ISSN: 3060-4745 IF(Impact Factor)10.41 / 2024
Volume 2, Issue 1

DEVELOPMENT OF A PROGRAM FOR PROCESSING 3D MODELS OF OBJECTS IN A COLLABORATIVE ROBOT WORKSPACE USING AN HD CAMERA

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

This article presents the research of methods for program development for processing 3D models of objects in a collaborative robot workspace using an HD camera. The features of creating and processing a point cloud in real time under different lighting conditions and image refresh rates are considered, and the influence of these factors on the accuracy of object recognition is also assessed. The presented experimental results show the optimal settings to ensure stable operation of the system in various production conditions. The proposed methodology allows to increase the accuracy and speed of the manipulator operation due to improved visual processing of images and 3D models.

Keywords: Industry 5.0, 3D Model Processing, Collaborative Robot, Manipulator, Hd Camera, Point Cloud, Object Recognition.

Introduction

In today's industrial environment, which is increasingly oriented towards Industry 5.0, the interaction between humans and robots is becoming particularly relevant. Collaborative robot manipulators, or cobots, perform tasks that were previously only available to humans, providing high accuracy, flexibility and speed in production. One of the main conditions for their effective operation is the ability to accurately process and interpret data about the environment, especially in the form of three-dimensional models of objects located in the robot's working area [1]-[6]. Processing 3D models allows the robot to quickly and correctly adapt to changes in the environment, avoid obstacles and interact with various objects, which significantly increases its safety and efficiency. Various methods and approaches can be used here

[7]-[25]. In particular, the availability of accurate algorithms for analyzing and processing such models is key to the successful use of robots in various industries, from the automotive and electronics industries to logistics and medicine.

At the same time, the development of effective algorithms for processing 3D data is a challenge due to the large amounts of data, the complexity of calculations and the need for high accuracy for successful recognition and classification of objects [26]-[30]. The use of modern methods of machine learning and computer vision, in particular deep neural networks, opens up new possibilities for processing three-dimensional models [31]-[42]. The combination of such technologies with libraries, such as Open3D and Point Cloud Library (PCL), allows you to create tools for detailed scanning and analysis of objects, which reduces the risk of errors and increases the speed of decision-making. In this context, research and development of algorithms for processing 3D models of objects in the working area of cobots are important for improving autonomous solutions and improving the quality of human-robot cooperation. The growing demand for robotic systems that can interact with the environment in real time and work safely next to people makes it necessary to conduct such research. Therefore, the development of algorithms that optimize the processing and recognition of objects in the robot space is a relevant topic for scientific research and practical applications in the field of new generation industrial and service robots.

Related works

Detection and recognition of objects is becoming especially relevant due to the increasing implementation of Industry 5.0 principles. Naturally, many scientists are working on this problem and dedicating their works to it. Let's look at several recent such works.

Let us begin with the work [43] where the authors present an overview of Computer Vision use in the Fifth Industrial Revolution. They also identify and present the changes between the two consequent periods: Industry 4.0 and Industry 5.0.

Rane, N. in [44] note that real-time object detection technologies, notably You Only Look Once (YOLO) and Faster Region Convolutional Neural Network (Faster R-CNN) algorithms are central for object detection. But the author explores challenges such as data privacy concerns, computational complexity, and ethical considerations. hese technologies are shaping a smarter, more connected, and efficient future across diverse sectors.

In their turn, Wang, H., and co-authors in [45] propose a reasoning approach towards factory unsafe states based on Digital Twin. First, a machine-readable semantic reasoning framework is introduced. Second, the ontology of unsafe states during production is modeled. Then, a high-fidelity virtual Digital Twin Workshop is constructed, which can simulate various workshop unsafe states and generate a virtual dataset.

Industry 5.0 focuses on collaboration between humans and machines, demanding robustness and efficiency and the accuracy of intelligent and innovative components used [46]. The paper [46] note that multimodal co-learning is one such approach to study the robustness of sensor fusion for missing and noisy modalities.

The paper [47] note that Cyber-Physical Human-centered Systems (CPHSs) have emerged to leverage operator capabilities in order to meet the goals of complex manufacturing systems towards human-centricity, resilience and sustainability. Such a CPHS enables increased operator safety and operation tracking in manufacturing processes that rely on collaborative robots and heavy machinery.

Researchers in [48] present an end-to-end vision system for bin-picking applications at the edge of industrial premises via industrial computers, where bin-picking refers to the dedicated selection and displacement of objects into designated areas. Our system is designed for human-robot collaboration in a range of distinct challenging industrial environments with a focus on automatically learning new objects.

So we see that the problems arising during the implementation of the principles of Industry 5.0 are quite diverse and attract many researchers. Further in this work we will present our solution to the problem of ensuring safety, namely, the definition of objects in the workspace of a collaborative robot.

Mathematical representation of image processing for object recognition and point cloud construction with 3D model visualization

To develop an algorithm for processing 3D models of objects in a collaborative robot workspace, at the first stage it is necessary to describe the sequence of the main stages:

– reading a frame from the camera;
– image processing for object recognition;
– building a point cloud;
– visualization of a 3D model.

Let's consider each of these stages in the form of mathematical models. The camera captures frames in the form of a pixel matrix, which is fed to the program input as:

$$I = H \times W \times C \quad (1)$$

I – image;

H – image height;

W – image width;

C – number of color channels (in the case of a color image C=3).

For image processing, we will use the MobileNetV2 model, which is loaded using TensorFlow, designed to classify objects in a frame. Before passing the image to the model, it is scaled to size , which is the standard input image size for MobileNetV2.

Then the image preprocessing, that is, when the image is normalized and scaled to values from -1 to 1, which is the standard normalization for the MobileNetV2 model, can be described by the following representation:

$$I_{norm} = \frac{I - 127.5}{127.5} \quad (2)$$

I_{norm} – normalized image;

I – original image;

127.5 – this is the average value of the range $[0, 255]$.

The value 127.5 in image preprocessing is often used in neural networks, particularly when preparing input images for deep learning. This value is used as part of the normalization process of pixel values to standardize the data before feeding it to the neural network, which helps the network learn faster and more accurately.

The normalized image is fed into a model $f_{MobilNet}$ that performs class prediction. The model returns a probability vector for each class:

$$\bar{y} = f_{MobilNet}(I_{norm}) \quad (3)$$

$\bar{y} \in R^K$ – class probability vector;

K – number of classes, for each of which the model returns the probability that the image belongs to this class.

To create a 3D model, the program builds a point cloud, which is simulated based on the intensity of the pixels of the frame. In real conditions, a special camera (for example, an RGB-D camera) is used to read the depth of each point. To calculate the spatial coordinates, it is proposed to use a fixed focal length of the camera f_x and f_y

(for the x and y axes). For each pixel (u,υ) with depth z, the spatial coordinates are calculated as follows:

$$x = \frac{(u - c_x) \cdot z}{f_x} \quad (4)$$

$$y = \frac{(\upsilon - c_y) \cdot z}{f_y} \quad (5)$$

$$z = d(u,\upsilon) \quad (6)$$

(u,υ) – pixel coordinates in the image;

(c_x, c_y) – coordinates of the center point of the image (camera center);

f_x та f_y – focal length along the axes x and y;

$d(u,\upsilon)$ – depth value for a pixel (u,υ), which is simulated through normalized gray values.

A point cloud is a collection of three-dimensional points that represent the spatial coordinates (x, y, z) of an object or scene. Typically, a point cloud is created by scanning the object using 3D scanners, lidar, photogrammetry, stereo cameras, or other technologies. Each point in the cloud corresponds to a specific location in space, and sometimes also has additional attributes, such as color or reflection intensity. In this case, a point cloud (P) is a set of spatial points:

$$P = \{(x_i, y_i, z_i) \mid i = 1, 2, ..., N\} \quad (7)$$

N – the number of points obtained from the image.

Visualization of a 3D model of objects in a collaborative robot workspace is the visualization of a cloud of points in space, that is, a visualization object is created and a cloud of points (P) is added to it, which is updated with each frame iteration.

Software implementation of the program for processing 3D models of objects in a collaborative robot workspace

The choice of the Python programming language for implementing a program for processing 3D object models in the workspace of a collaborative robot manipulator is optimal due to its powerful libraries and ease of integration with modern computer vision and machine learning tools. Python has a large set of ready-made libraries, such as OpenCV for image processing, Open3D for working with 3D geometry, and TensorFlow for deep learning, which allows you to significantly simplify the process of creating complex object recognition and analysis algorithms. Due to the simplicity of the syntax and high readability of the code, Python is an ideal choice for developers working with artificial intelligence algorithms and 3D data processing, as it allows you to quickly prototype and test solutions. Python's high compatibility with various hardware platforms and operating systems facilitates easy integration of software for robotic systems in real conditions.

Let us present a description of the software implementation of the mathematical representation of image processing for object recognition and point cloud construction with 3D model visualization.

```
def predict_objects(frame):
    img = cv2.resize(frame, (224, 224))
    img = tf.keras.applications.mobilenet_v2.preprocess_input(img)
    img = np.expand_dims(img, axis=0)
    predictions = model.predict(img)
    decoded_predictions =
tf.keras.applications.mobilenet_v2.decode_predictions(predictions, top=3)
    return decoded_predictions[0]
```

This code snippet is used to recognize objects in an image captured by a camera. First, the frame is scaled to 224x224, which is a requirement of the MobileNetV2 model, and then the data is processed to match the model. The `model.predict(img)` function makes a prediction for the objects in the frame, and then `decode_predictions` decodes the results, returning a list of the three most likely objects and their probabilities.

```
predictions = predict_objects(frame)
    for i, (imagenet_id, label, score) in enumerate(predictions):
        text = f"{label}: {score:.2f}"
        cv2.putText(frame, text, (10, 30 + i * 30),
cv2.FONT_HERSHEY_SIMPLEX, 1, (255, 0, 0), 2)
```

This code snippet displays the results of object recognition on a video frame in real time. Specifically, it gets object predictions from the `predict_objects(frame)` function, and then for each recognized object, writes its label (`label`) and probability (`score`) to the video frame. Text is applied to the image using `cv2.putText` to visually label the identified objects along with their probabilities at the given positions.

```
depth_image = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
depth_image = cv2.resize(depth_image, (640, 480))
depth_image = np.float32(depth_image) / 255.0
```

This code snippet processes a video frame, converting it to grayscale to simulate image depth. First, `frame` is converted to black and white using `cv2.COLOR_BGR2GRAY`, then resized to a standard 640x480 pixels. Finally, the pixel values are normalized to a range of 0 to 1, which makes it easier to perform further calculations to simulate the point cloud.pcd = o3d.geometry.PointCloud()

```
h, w = depth_image.shape
fx, fy = 500, 500  # Simulating focal lengths
cx, cy = w // 2, h // 2
points = []
for v in range(h):
    for u in range(w):
        z = depth_image[v, u]
        x = (u - cx) * z / fx
        y = (v - cy) * z / fy
        points.append((x, y, z))
pcd.points = o3d.utility.Vector3dVector(np.array(points))
```

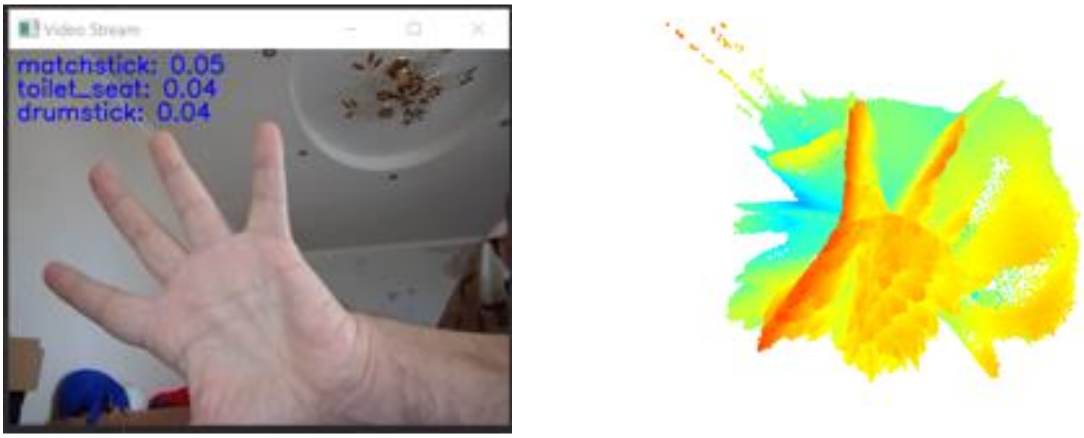
This code snippet creates a point cloud object, `pcd`, to display a 3D structure based on a depth image `depth_image`. Each pixel in the image is converted to a 3D coordinate (x, y, z) using the focal lengths `fx` and `fy` and the image center `(cx, cy)` to calculate the spatial location of each point. The resulting coordinates are added to a list `points`, which is then converted to an Open3D-compatible format and assigned to `pcd.points`.vis.clear_geometries()

```
vis.add_geometry(pcd)
vis.poll_events()
vis.update_renderer()
```

This code snippet updates the point cloud visualization in the Open3D window. It first clears the previous geometry using `vis.clear_geometries()`, and then adds a new point cloud `pcd` using `vis.add_geometry(pcd)`. The `vis.poll_events()` and `vis.update_renderer()` calls process the events and update the visualization to reflect the current changes in real time.

An example of the program processing 3D models of objects in collaborative robot workspace based on an HD camera is shown in Figure 1.

ISSN: 3060-4745 IF(Impact Factor)10.41 / 2024
Volume 2, Issue 1



a) b)

a) «Video Stream» window

b) «3D Object Models» window

Figure 1: Example of the program processing 3D models of objects in a collaborative robot workspace based on an HD camera

Let us conduct a series of experiments to evaluate the performance of the program for processing 3D models of objects in a collaborative robot workspace at different settings of the video stream frame rate. We investigate how the program processes data from the camera in real time under different settings and conditions, for example, by changing the frame rate of the video stream. This will allow us to determine the delays in displaying data and visualizing the 3D model and to assess how quickly the system updates the image when the load increases. In the conclusions, present a table with the data. Table 1 shows the values of the frame rate (FPS), the average frame processing time, the image update delay and the point cloud update rate in real time.

Table 1: The first experiment results

Frame rate (FPS)	Average frame processing time (ms)	Image refresh delay (ms)	Point cloud refresh rate (FPS)
15	53	72	10
20	62	87	13
25	77	104	17
30	92	123	20
35	111	141	22

Acumen: International Journal of Multidisciplinary Research
ISSN: 3060-4745 IF(Impact Factor)10.41 / 2024
Volume 2, Issue 1

40	136	164	24
45	154	186	25
50	176	203	27

Let us present the obtained data of the first experiment in the form of a combined graph in Figure 2.

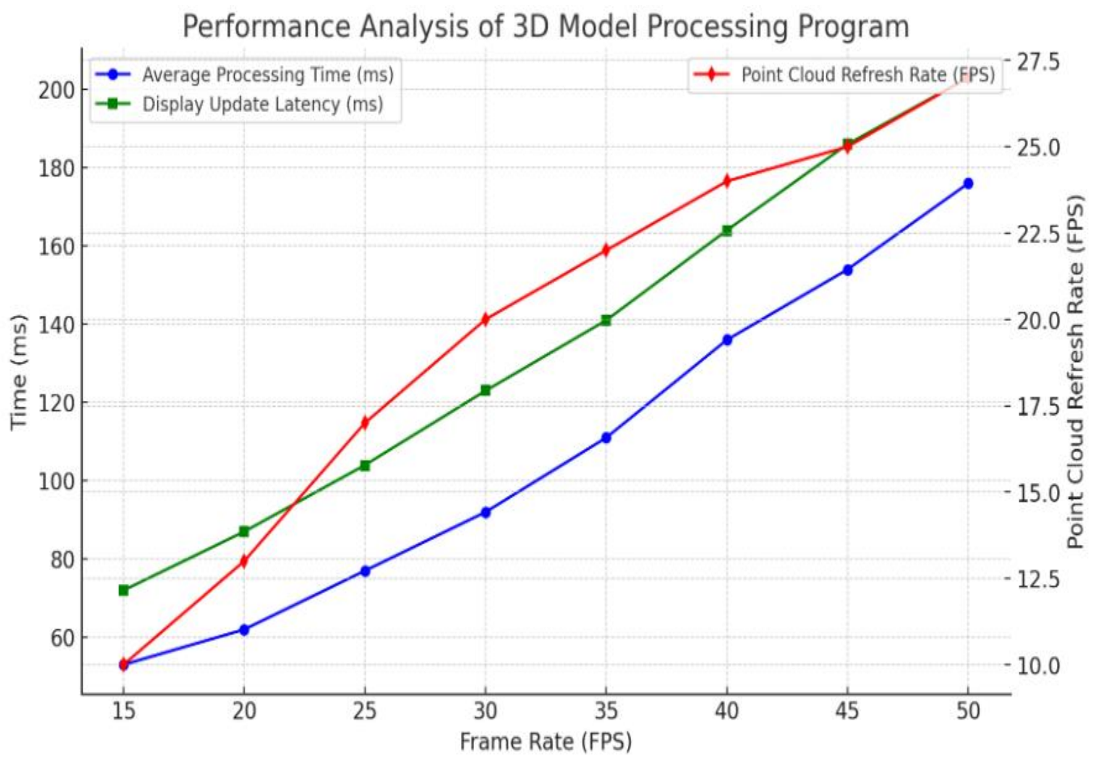


Figure 2: Combined graph of 3D modeling program performance analysis

The data from the first experiment allows us to assess the impact of frame rate on the performance and ability of the program to process 3D models in the collaborative robot workspace. At a low frame rate of 15 frames per second (FPS), the system demonstrates an average frame processing time of 53 ms and an image refresh delay of 72 ms, and the point cloud refresh rate is 10 FPS. With an increase in frame rate to 20 FPS, the frame processing time increases to 62 ms and the delay to 87 ms, indicating an increased load on the system. At a rate of 25 FPS, frame processing increases to 77 ms and the delay to 104 ms, indicating that an increase in frame rate requires additional computing resources and affects the refresh rate.

At 30 FPS, the processing time increases to 92 ms and the delay reaches 123 ms, which is repeated at each subsequent increase: frame processing reaches 111 ms at 35 FPS and 136 ms at 40 FPS, and the delay increases to 141 ms and 164 ms, respectively. At a high frame rate of 50 FPS, the system processes frames in 176 ms with an update latency of 203 ms, indicating some limitations for real-time processing. The point cloud update rate also increases with frame rate, reaching 27 FPS at the maximum input stream rate of 50 FPS, indicating the system's ability to update the 3D visualization faster as the amount of data increases.

The second experiment is associated with different lighting conditions, which makes it possible to check how the results of recognition and point cloud construction change when the illumination level changes. The results obtained during the second experiment are given in Table 2.

Table 2: The second experiments results

Lighting level (lux)	Point cloud quality (%)	Object recognition accuracy (%)	Frame processing time (ms)	Point cloud update delay (ms)	Notes
100	76	66	83	112	Low light, strong shadows
300	82	78	73	94	Normal daylight, weak shadows
500	91	87	65	83	Moderate artificial lighting

700	96	89	62	81	Bright artificial lighting, no shadows
1000	94	87	64	82	Very bright lighting, slight overexposure
1500	71	63	92	107	Excessive lighting, overexposure and glare

Let us present the obtained data of the second experiment in the form of a combined graph in Figure 3.

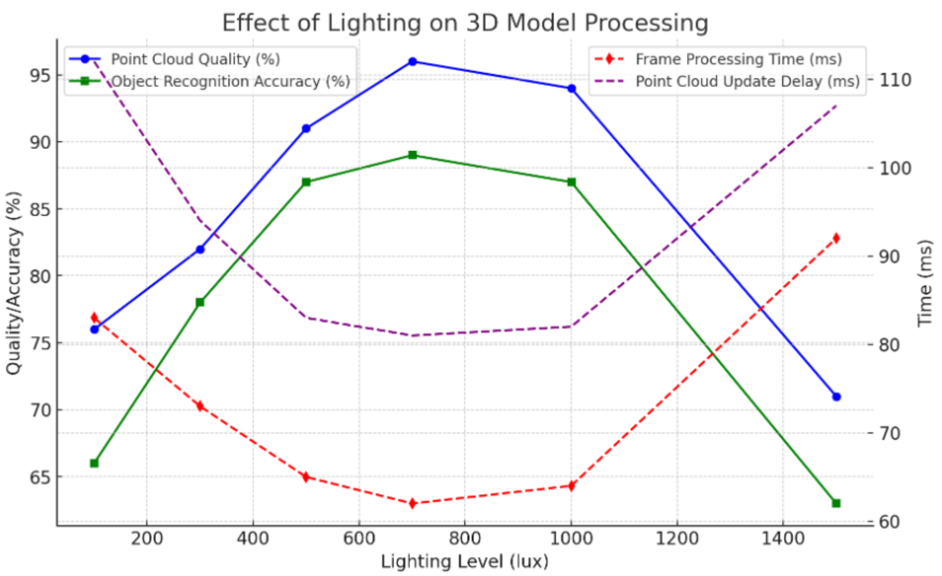


Figure 3: Combined graph of the impact of lighting on 3D model processing

Analysis of the obtained experimental data shows a significant influence of the illumination level on the quality of the point cloud and the accuracy of object recognition in the collaborative robot workspace. At low illumination (100 lux), the quality of the point cloud is 76%, and the accuracy of object recognition is 66%, which indicates a decrease in quality due to the presence of shadows. With an increase in illumination to 300 lux, the quality of the point cloud and the accuracy of recognition increase to 82% and 78%, respectively, and the frame processing time decreases to 73 ms, which indicates an improvement in the conditions for processing. The optimal level for processing is the illumination level within 500-700 lux, where the quality of the point cloud reaches 91-96%, the accuracy of recognition is 87-89%, and the frame processing time decreases to 62 ms. This is explained by the stability of the lighting conditions and the absence of shadows. Overillumination conditions (1000 lux) show a slight decrease in point cloud quality to 94%, while accuracy remains stable at 87%, although slight overillumination appears. Under excessive illumination (1500 lux), point cloud quality drops sharply to 71%, and recognition accuracy drops to 63% due to the influence of glare, which significantly degrades the system's ability to accurately recognize objects.

Conclusion

The article considered the process of developing software for processing 3D models of objects in a collaborative robot workspace using an HD camera. Based on modern methods of computer vision and neural networks, it was possible to create a program that is capable of recognizing and visualizing objects in real time, using the MobileNetV2 architecture for accurate and fast recognition. The aspects of integrating 3D visualization into the robot's working area to build accurate point clouds were also considered, which allows forming three-dimensional models of objects and spatially assessing their positions, which is especially important for collaborative environments. The use of OpenCV and Open3D libraries ensured fast processing of the video stream and convenient work with 3D data. The program allows robot-manipulators to work more efficiently, reducing the risk of collisions and increasing safety due to the accurate determination of the location of objects and the ability to recognize them in dynamic conditions. This also opens up the possibility for further automation in industrial applications, where the presence of an accurate 3D picture of the workspace can optimize the performance of tasks by manipulators. The work demonstrates that the further development of such software systems has great potential in the field of robotics, especially in Industry 5.0, where accuracy and integration with artificial intelligence are becoming critically important. The developed software solution can become the basis for complex robotic systems that require advanced visual control and analysis capabilities.

References

1. Abu-Jassar, A., & et al. (2023). Obstacle Avoidance Sensors: A Brief Overview. Multidisciplinary Journal of Science and Technology, 3(5), 4-10.
2. Yevsieiev, V., & et al. (2024). Research of Existing Methods of Representing a Collaborative Robot-Manipulator Environment within the Framework of Cyber-Physical Production Systems. Multidisciplinary Journal of Science and Technology, 4(9), 112-120.
3. Maksymova, S., & et al. (2024). Comparative Analysis of methods for Predicting the Trajectory of Object Movement in a Collaborative Robot-Manipulator Working Area. Multidisciplinary Journal of Science and Technology, 4(10), 38-48.
4. Samoilenko, H., & et al. (2024). Review for Collective Problem-Solving by a Group of Robots. Journal of Universal Science Research, 2(6), 7-16.
5. Yevsieiev, V., & et al. (2024). Human Operator Identification in a Collaborative Robot Workspace within the Industry 5.0 Concept. Multidisciplinary Journal of Science and Technology, 4(9), 95-105.
6. Nevliudov, I., & et al. (2023). A Small-Sized Robot Prototype Development Using 3D Printing. CAD In Machinery Design Implementation and Educational Issues (CADMD'2023) : proceedings of the XXXI International Conference, Suprasl, 12.
7. 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.
8. 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.
9. Matarneh, R., Maksymova, S., Deineko, Z., & Lyashenko, V. (2017). Building robot voice control training methodology using artificial neural net. International Journal of Civil Engineering and Technology, 8(10), 523-532.
10. 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.
11. Sotnik, S., Matarneh, R., & Lyashenko, V. (2017). System model tooling for injection molding. International Journal of Mechanical Engineering and Technology, 8(9), 378-390.
12. 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.

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

14.Lyashenko, V. V., Babker, A. M. A. A., & Kobylin, O. A. (2016). The methodology of wavelet analysis as a tool for cytology preparations image processing. Cukurova Medical Journal, 41(3), 453-463.

15.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.

16.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.

17.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.

18.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.

19.Ahmad, M. A., Baker, J. H., Tvoroshenko, I., & Lyashenko, V. (2019). Modeling the structure of intellectual means of decision-making using a system-oriented NFO approach. International Journal of Emerging Trends in Engineering Research, 7(11), 460-465.

20.Lyashenko, V., Kobylin, O., & Selevko, O. (2020). Wavelet analysis and contrast modification in the study of cell structures images. International Journal of Advanced Trends in Computer Science and Engineering, 9(4), 4701-4706.

21.Lyashenko, V., & et al.. (2021). Wavelet ideology as a universal tool for data processing and analysis: some application examples. International Journal of Academic Information Systems Research (IJAISR), 5(9), 25-30.

22.Ahmad, M. A., Baker, J. H., Tvoroshenko, I., Kochura, L., & Lyashenko, V. (2020). Interactive Geoinformation Three-Dimensional Model of a Landscape Park Using Geoinformatics Tools. International Journal on Advanced Science, Engineering and Information Technology, 10(5), 2005-2013.

23.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.

24. Babker, A. M., Abd Elgadir, A. A., Tvoroshenko, I., & Lyashenko, V. (2019). Information technologies of the processing of the spaces of the states of a complex biophysical object in the intellectual medical system health. International Journal of Advanced Trends in Computer Science and Engineering, 8(6), 3221-3227.
25. Khan, A., Joshi, S., Ahmad, M. A., & Lyashenko, V. (2015). Some effect of Chemical treatment by Ferric Nitrate salts on the structure and morphology of Coir Fibre Composites. Advances in Materials Physics and Chemistry, 5(1), 39-45.
26. Gurin, D., & et al. (2024). Using Convolutional Neural Networks to Analyze and Detect Key Points of Objects in Image. Multidisciplinary Journal of Science and Technology, 4(9), 5-15.
27. Abu-Jassar, A., & et al. (2024). The Optical Flow Method and Graham's Algorithm Implementation Features for Searching for the Object Contour in the Mobile Robot's Workspace. Journal of Universal Science Research, 2(3), 64-75.
28. 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.
29. Chala, O., & et al. (2024). Switching Module Basic Concept. Multidisciplinary Journal of Science and Technology, 4(7), 87-94.
30. 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.
31. 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.
32. 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.
33. 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.

34. 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.
35. Tahseen A. J. A., & et al.. (2023). Binarization Methods in Multimedia Systems when Recognizing License Plates of Cars. International Journal of Academic Engineering Research (IJAER), 7(2), 1-9.
36. Orobinskyi, P., Petrenko, D., & Lyashenko, V. (2019, February). Novel approach to computer-aided detection of lung nodules of difficult location with use of multifactorial models and deep neural networks. In 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM) (pp. 1-5). IEEE.
37. Matarneh, R., Sotnik, S., Belova, N., & Lyashenko, V. (2018). Automated modeling of shaft leading elements in the rear axle gear. International Journal of Engineering and Technology (UAE), 7(3), 1468-1473.
38. Abu-Jassar, A. T., Attar, H., Amer, A., Lyashenko, V., Yevsieiev, V., & Solyman, A. (2024). Remote Monitoring System of Patient Status in Social IoT Environments Using Amazon Web Services (AWS) Technologies and Smart Health Care. International Journal of Crowd Science, 8.
39. Lyubchenko, V., Veretelnyk, K., Kots, P., & Lyashenko, V. (2024). Digital image segmentation procedure as an example of an NP-problem. Multidisciplinary Journal of Science and Technology, 4(4), 170-177.
40. Babker, A. M., Suliman, R. S., Elshaikh, R. H., Boboyorov, S., & Lyashenko, V. (2024). Sequence of Simple Digital Technologies for Detection of Platelets in Medical Images. Biomedical and Pharmacology Journal, 17(1), 141-152.
41. Yevstratov, M., Lyubchenko, V., Amer, A. J., & Lyashenko, V. (2024). Color correction of the input image as an element of improving the quality of its visualization. Technical science research in Uzbekistan, 2(4), 79-88.
42. Attar, H., Abu-Jassar, A. T., Lyashenko, V., Al-qerem, A., Sotnik, S., Alharbi, N., & Solyman, A. A. (2023). Proposed synchronous electric motor simulation with built-in permanent magnets for robotic systems. SN Applied Sciences, 5(6), 160.
43. Tzampazaki, M., & et al. (2024). Machine Vision—Moving from Industry 4.0 to Industry 5.0. Applied Sciences, 14(4), 1471.
44. Rane, N. (2023). YOLO and Faster R-CNN object detection for smart Industry 4.0 and Industry 5.0: applications, challenges, and opportunities. Available at SSRN 4624206.

45. Wang, H., & et al. (2023). A safety management approach for Industry 5.0′ s human-centered manufacturing based on digital twin. Journal of Manufacturing Systems, 66, 1-12.
46. Rahate, A., & et al. (2023). Employing multimodal co-learning to evaluate the robustness of sensor fusion for industry 5.0 tasks. Soft Computing, 27(7), 4139-4155.
47. Fraga-Lamas, P., & et al. (2022). Mist and edge computing cyber-physical human-centered systems for industry 5.0: A cost-effective IoT thermal imaging safety system. Sensors, 22(21), 8500.
48. Wagner, R., & et al. (2023). IndustrialEdgeML-End-to-end edge-based computer vision systemfor Industry 5.0. Procedia Computer Science, 217, 594-603.