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

DEVELOPMENT OF A MODEL FOR RECOGNIZING VARIOUS OBJECTS AND TOOLS IN A COLLABORATIVE ROBOT WORKSPACE

Vladyslav Yevsieiev[1], Amer Abu-Jassar[2], Svitlana Maksymova[1], Nataliia Demska[1]

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

[2]Department of Computer Science, College of Information Technology, Amman Arab University, Amman, Jordan

Abstract

The article discusses the development of a model for recognizing objects and tools in the robot's workspace, which is based on computer vision and machine learning methods to ensure safe interaction within the framework of Industry 5.0. The model allows increasing the accuracy and reliability of object recognition in complex conditions, adapting robots to changing tasks. The results can be used for integration into robotic platforms operating in flexible manufacturing environments, ensuring flexible automation and a human-centric approach.

Keywords: Object Recognition, Robotic Systems, Computer Vision, Machine Learning, Robot Workspace, Industry 5.0.

Introduction

In the context of the rapid development of Industry 5.0, which is aimed at the harmonious coexistence of humans and technologies, robotic systems are taking on new roles, in particular in the field of safe and effective cooperation with people in production [1]-[4]. Modern robotic platforms are no longer limited to automating routine processes, but are beginning to perform more complex tasks that require adaptation to rapidly changing conditions and accurate recognition of surrounding objects [5]-[30]. Various methods and approaches can be used here [31]-[41].

Of particular importance is the ability of robots to distinguish between different tools and objects in the work area, which allows them to make operational decisions about interacting with them based on their types and characteristics. This opens up opportunities for flexible adaptation of the robot to new tasks, automation of complex technological processes and increased safety in the workplace. However, existing object recognition technologies often have low accuracy when processing complex or noisy data, which requires the development of new methods to achieve high reliability. Within the framework of the Industry 5.0 concept, a robot must be able not only to recognize objects, but also to understand the context of tool use and act accordingly to changing tasks. Such adaptability is achieved by combining artificial intelligence, computer vision and machine learning algorithms, which together create new possibilities for robotic systems. This research aims to develop a mathematical model that will allow robots to effectively identify and classify objects in the work area, ensuring their reliable interaction with the environment. The successful implementation of such a model will contribute to the development of robotic platforms that can support autonomous interaction with tools and objects, automatically choosing the optimal paths to perform production tasks. In addition, such developments meet the principles of human-centricity and sustainable development, which are key in the Industry 5.0 concept. As a result, robotic systems will be able to provide higher productivity, minimize risks to workers, and expand opportunities for integration into new industries.

Related works

In the modern world, many scientists are engaged in the implementation of the principles of the Industry 5.0 concept. They consider a wide variety of problems that arise when solving the above-mentioned task. Let us consider several such works.

First of all, let us analyze the work [42]. There is noted the critical role and implications of Cobotics in the context of Industry 5.0. The study addresses the research problem of effectively integrating Cobots into industrial processes, considering technical, economic, and social challenges.

Collaborative robotics, or “cobotics”, is a major enabling technology of Industry 5.0, which aspires at improving human dexterity by elevating robots to extensions of human capabilities and, ultimately, even as team members. This fact is noted in [43].

A review [44] was performed aimed at investigating the effect of robot design features on their human counterparts. Its results showcased the many to many relationships between robot design features and effects on operators.

Doyle Kent, M., & Kopacek, P. in [45] arise next questions whether it is possible to ensure that humans have a place in the highly automated workplace of the future

(Industry 5.0) by optimizing human capital; and whether it is possible for traditional educational provider supply the skills required to educate this modern worker or do we require an innovative educational system?

Prassida, G. F., & Asfari, U. in [46] provide a holistic view of the acceptance of collaborative robots (cobots) in the manufacturing context by adopting the socio-technical perspective to the Industry 5.0 era. Grounding on the Unified Theory of Acceptance and Use of Technology (UTAUT) and Socio-Technical Systems theory (STS), this study proposes a conceptual model to better understanding critical factors that influence the acceptance of cobots and how these factors can drive perceived work performance improvement in the organizational level.

The scientists in [47] note that one relevant the most relevant challenges of Industry 5.0 is the design of human-centered smart environments (i.e., that prioritize human well-being while maintaining production performance).

Thus, we see a variety of issues arising during the implementation of Industry 5.0 technology. Our vision of a possible solution to the problem of recognizing objects and tools is presented further in this article.

Mathematical model of various objects and tools recognition in a collaborative robot workspace for making decisions about further actions

To create a mathematical model for recognizing various objects and tools in the robot's working area and making decisions for interacting with them, a model based on neural networks and image processing methods is used. As part of these studies, the following mathematical model is proposed, which covers the main stages: processing input data, classifying objects, determining position, calculating interaction parameters and making decisions.

The first stage: The robot perceives the working area using sensors or a camera that provide an image or a three-dimensional map. Let's assume that the image has a resolution $W \times H$ (width and height) and is represented as a set of pixels. We make the following variables $I(x, y)$ – intensity or color value for a pixel with coordinates (x, y) and $D(x, y)$ – depth or distance to the object in the working area (for stereo images or using LiDAR). Before starting recognition, smoothing, color normalization, noise filtering and contrast equalization methods are applied.

The next step in the input data processing stage is the separation of objects using segmentation. Segmentation consists of dividing an image into parts to highlight areas that may be objects or tools.

To separate objects based on color or depth, it is proposed to use the threshold segmentation method, which can be described by the following model:

$$S(x,y)=\begin{cases}1,\text{if } I(x,y)\in[I_{\min},I_{ma[}]\text{and } D(x,y)\in[D_{\min},D_{\max}]\\ 0,\ \text{otherwise}\end{cases} \tag{1}$$

$S(x,y)$ – segmentation mask;

$I_{\min}$ and $I_{\max}$ – intensity range for segmentation;

$D_{\min}$ and $D_{\max}$ – range for depth.

The convolution method will be used to calculate image gradients to determine the contours of objects, which can be represented by the following expression:

$$G(x,y)=\sum_{i=-k}^{k}\sum_{j=-k}^{k} I(x+i,y+j)\cdot K(i,j) \tag{2}$$

$G(x,y)$ – result of convolution;

$K(i,j)$ – convolution kernel (e.g., Sobel operator or other solution).

The second stage is object classification using a neural network. To classify objects, a convolutional neural network (CNN) is used, which is trained on data containing images and object labels. The CNN model can be rearranged as follows:

– input layer:

$$X=\{I(x,y)\}_{x=1,y=1}^{W,H} \tag{3}$$

X – input layer.

– convolutional layer:

$$F_l=\sigma\left(\sum_{i=1}^{N}\sum_{j=1}^{N}K_{ij}\cdot X_{(x+i,y+j)}+b\right) \tag{4}$$

F_l – filtered image on l-th layer;

σ – activation function (e.g.ReLU);

K – convolution kernels in CNN;

b – displaced.

– max pooling layer for dimensionality reduction:

$$P(x, y) = \max(X_{ij}),\ (i, j) \in \text{area}\ . \quad (5)$$

– fully connected layer for object class output:

$$y = f(W \cdot P + b)\,, \quad (6)$$

y – probability vector for each class;

W – weight matrix;

b – bias;

f – a nonlinear activation function that converts logits (values from the intermediate layer) into probabilities.

The third stage is to determine the position of objects in the collaborative robot workspace. If objects are recognized, their positions are determined taking into account the depth and pixel coordinates, and can be described as follows:

$$(x_{ob}, y_{ob}, d_{ob}) = O(S(x, y) \cdot D(x, y)) \quad (7)$$

(x_{ob}, y_{ob}, d_{ob}) – coordinates and depth of the center of mass (or center of gravity) of the object, defined in the robot's working area;

O – operator that calculates the coordinates of the center of mass of an object by weighting the depth values according to the segmentation mask. This operator calculates the average value of the coordinates (x, y) taking into account the mask $S(x, y)$ and the depth $D(x, y)$ to obtain the position of the center of mass of the object in the image and in space;

$S(x, y)$ – a two-dimensional image segmentation function that determines whether each point with coordinates (x, y) belongs to an object;

$D(x, y)$ – a function that represents the depth or distance to each point (x, y) in an image.

And there is the last stage of decision-making for interaction with objects in the collaborative robot workspace. Based on the recognized objects and tools, the robot makes a decision, in particular, selects appropriate actions:

– the decision to capture the object, can be represented by the following expression:

$$R = \begin{cases} \text{"capture", if } d_{ob} < d_{capture} \text{ and } C_{ob} = \text{"tool"} \\ \text{"go around", if } C_{ob} = \text{"obstacle"} \\ \text{"continue", otherwise} \end{cases} \tag{8}$$

C_{ob} – object class (tool, obstacle, etc.).

– planning a trajectory to avoid or approach an object:

$$T(x, y) = f(x, y, x_{ob}, y_{ob}) \tag{9}$$

$T(x, y)$ – trajectory built for interaction;

f – scheduling algorithm (e.g., A* or D*).

Taking into account all stages, the mathematical model of decision-making based on object recognition is described by the expression:

$$N(I, D) = R(CNN(S(I, D)), (x_{ob}, y_{ob}, d_{ob})) \tag{10}$$

N – decision-making function;

$S(I, D)$ – segmentation result;

CNN – classification function;

R – solution for interacting with the object.

The developed general mathematical model of decision-making based on object recognition $N(I, D)$ provides a number of advantages for the tasks of object recognition in the robot's workspace. The use of a convolutional neural network (CNN) in combination with segmentation $S(I, D)$, which takes into account both image intensity I and depth data D, allows the model to better distinguish objects, taking into account their three-dimensional structure and position. This provides increased accuracy and robustness in complex conditions, where objects may partially overlap or change their orientation. The parameters of the center of mass of the object (x_{ob}, y_{ob}, d_{ob}) allow the model to take into account the position and distance to objects, which facilitates decision-making regarding interaction with them in real space. The component R combines the processed features and positional parameters, creating a holistic approach to object recognition and response. Such a comprehensive model increases the efficiency and reliability of robotic systems in a changing production environment, which meets the requirements of Industry 5.0 and contributes to flexible automation and integration of human-centric technologies.

Software implementation of a program for recognizing various objects and tools in the a collaborative robot workspace

The choice of the Python programming language for developing an object recognition program in the collaborative robot workspace is justified by its powerful capabilities in the field of machine learning and computer vision, as well as the availability of numerous specialized libraries. Python has a simple and understandable syntax, which makes it convenient for rapid development and maintenance of code, especially in complex engineering projects. In combination with the TensorFlow library, Python provides extensive capabilities for working with neural networks, in particular convolutional (CNN), which are the basis for object recognition. Using TensorFlow also allows you to use pre-trained models, such as MobileNetV2, to speed up the development process and improve recognition accuracy. The integration of the OpenCV library, which has image processing functions such as noise filtering, smoothing, segmentation, and more, makes Python an ideal tool for processing video streams in real time. The NumPy library provides efficient work with multidimensional arrays, which is a key aspect when manipulating images and processing results, especially during segmentation and classification. In addition, Python is a cross-platform language, which allows you to run the program on different operating systems, providing flexibility and portability. Thanks to an active community of developers and a large number of open resources, Python offers stable support and continuous improvement of the tools necessary for the development of modern computer vision and artificial intelligence systems, which is critically important for collaborative robotics. Let us describe the software implementation of the recognition of various objects and tools in a collaborative robot workspace.

```
model = tf.saved_model.load(
    r"C:\Users\Vladyslav\.cache\kagglehub\models\tensorflow\ssd-mobilenet-v2\tensorFlow2\fpnlite-320x320\1")
```

This code snippet loads the pre-trained ssd-mobilenet-v2 model from the TensorFlow library, saved in the SavedModel format, from the specified path. The model is used to recognize objects in images or video streams, allowing the program to automatically determine object classes and their coordinates in the frame.

```
def preprocess_image(image):
    image = cv2.GaussianBlur(image, (5, 5), 0)
```

```
    image = cv2.normalize(image, None, 0, 255, cv2.NORM_MINMAX)
    return image
```

This code snippet defines the `preprocess_image` function, which performs preprocessing on the image to improve its quality before further analysis. The function applies smoothing using a Gaussian filter to reduce noise and normalizes the pixel intensity to a range of 0 to 255. This helps improve segmentation and object recognition results.

```
def threshold_segmentation(image, low_intensity, high_intensity):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    _, thresholded = cv2.threshold(gray, low_intensity, high_intensity, cv2.THRESH_BINARY)
    return thresholded
```

This code snippet defines the `threshold_segmentation` function, which performs threshold segmentation on an image to extract objects. First, the image is converted to grayscale, and then a binary threshold is applied, which turns pixels into black or white depending on their intensity. This simplifies further processing, making it easier to identify objects in the image.

```
def calculate_center_of_mass(box, frame_shape):
    y1, x1, y2, x2 = box
    height, width = frame_shape[:2]
    center_x = int((x1 + x2) / 2 * width)
    center_y = int((y1 + y2) / 2 * height)
    return center_x, center_y
```

This code snippet defines the function `calculate_center_of_mass`, which calculates the coordinates of the center of mass of an object using the coordinates of its bounding box `box`. Given the dimensions of the frame `frame_shape`, the function returns the center coordinates `center_x` and `center_y`, which helps to more accurately determine the position of the object for further interaction.

```
num_detections = int(detections['num_detections'][0])
    detection_classes
    detections['detection_classes'][0].numpy().astype(np.int64)
    detection_boxes = detections['detection_boxes'][0].numpy()
    detection_scores = detections['detection_scores'][0].numpy()
```

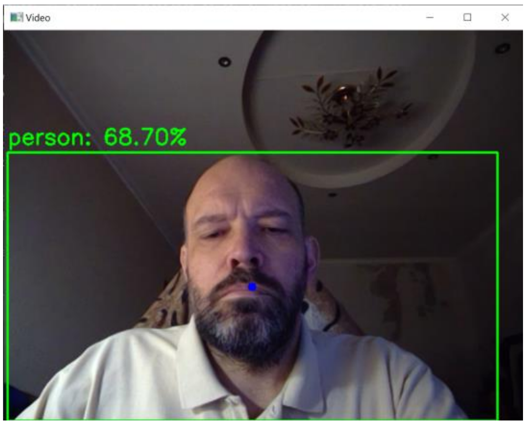
This code snippet extracts object recognition results from the model's detection output. `num_detections` specifies the number of objects found, and `detection_classes`, `detection_boxes`, and `detection_scores` contain the classes,

Acumen: International Journal of Multidisciplinary Research
ISSN: 3060-4745 IF(Impact Factor)10.41 / 2024
Volume 2, Issue 1
bounding box coordinates, and probabilities for each object, respectively. This allows the program to process and display information about the detected objects in the image.

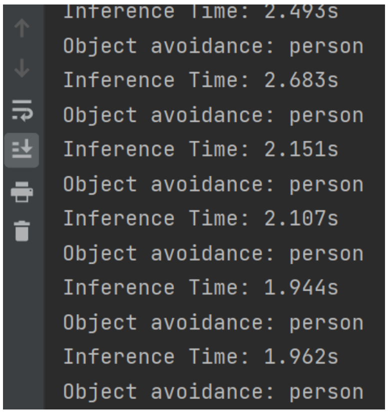
```
if class_name in ['person', 'car', 'bicycle']:
    print(f"Object avoidance: {class_name}")
  else:
    print(f"Capturing the object: {class_name}")
```

This code fragment checks the object class to determine whether it is an obstacle to avoid or an object to capture. If the object class belongs to the obstacle list (`person`, `car`, `bicycle`), avoidance is performed, otherwise — capture. This helps to decide on the robot's further actions depending on the type of object.

An example of the developed program for recognizing various objects and tools in the collaborative robot's workspace is shown in Figure 1.



a)



b)

a) recognition program window; b) decision terminal window.

Figure 1: An example of the work of the developed program for recognizing various objects and tools in a collaborative robot workspace

Let's conduct an experiment to test the developed program for recognizing various objects and tools in a collaborative robot workspace, for example, to check the model in situations where objects are partially covered or noise is superimposed on the image (for example, by varying the lighting or adding background noise). This will allow us to assess the model's resistance to real conditions, where errors may occur due to complex background conditions. The results obtained during the experiment are given in Table 1, and Figure 2 shows a graph

Table 1: Results in different conditions

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

Experimental conditions	Proportion of objects recognized (%)	Precision (%)	Recall (%)	F1-measure (%)
No noise, normal lighting	98%	97%	95%	96%
Partial overlap (25%)	85%	83%	78%	80%
Partial overlap (50%)	65%	60%	55%	57%
Low lighting	70%	68%	66%	67%
High lighting	80%	75%	70%	72%
Added background noise (10%)	90%	85%	83%	84%
Added background noise (20%)	75%	70%	65%	67%
Added background noise (30%)	60%	55%	50%	52%
Partial overlap + low light	50%	48%	45%	46%
Partial overlap + background noise	55%	53%	50%	51%

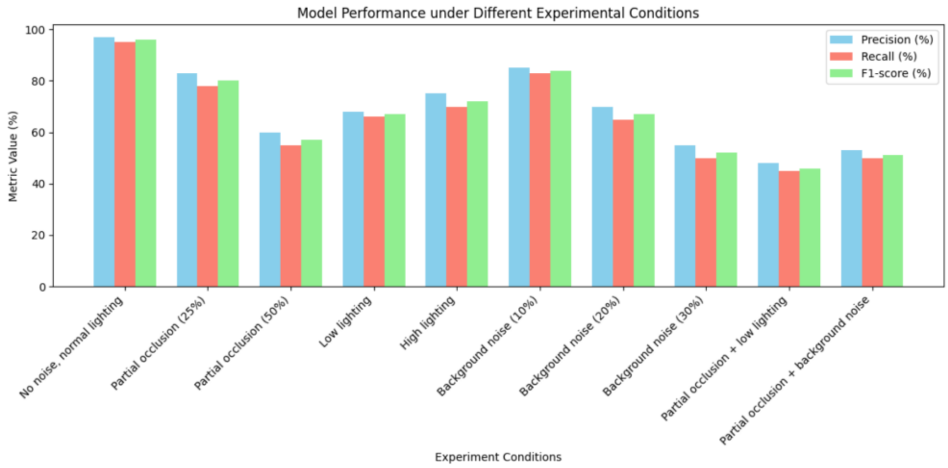


Figure 2: Model Performance under Different Experimental Conditions Graph

Analysis of the obtained experimental data shows that the object recognition model demonstrates high accuracy (Precision 97%) and completeness (Recall 95%) in standard conditions without interference, which confirms its basic adequacy. With partial overlap of objects, accuracy and completeness decrease (to 83% and 78%, respectively, at 25% overlap), and at 50% overlap these indicators fall even more, which indicates the vulnerability of the model to partial visibility of objects. In low-light conditions, Precision and Recall indicators decrease to 68% and 66%, while with additional background noise of 10% the model remains relatively stable (Precision 85%, Recall 83%). However, with an increase in noise to 30%, the indicators drop significantly, especially Recall to 50%, which indicates a decrease in the model's ability to correctly identify objects under strong interference. The F1-measures, which take into account both Precision and Recall, show a similar trend, confirming the general logic and stability of the model's quality degradation with increasing noise, decreasing illumination, or partial overlap. Overall, the model performs well under optimal conditions, but its robustness to noise and partial overlap needs improvement to ensure reliability in real-world conditions.

Conclusion

The article presents a developed model for recognizing various objects and tools in a collaborative robot workspace, which provides automatic determination of object classes and positions for safe and effective interaction with them. The model demonstrates high accuracy in standard conditions, however, experimental results indicate the need for improvement in conditions of low illumination, partial overlapping of objects and increased noise, where the recognition quality decreases. This indicates the importance of additional image processing mechanisms, such as adaptive segmentation, improved smoothing and methods that take into account the three-dimensional structure of the workspace. Further research prospects include expanding the functionality of the model using more complex neural networks, such as deep convolutional networks and transformers, which can improve noise immunity and ensure reliable operation in difficult conditions. Research can also focus on integrating data from additional sensors, such as LiDAR and ultrasonic sensors, for more accurate determination of the distance to objects. This will contribute to the creation of a comprehensive detection and classification system that takes into account not only visual characteristics, but also spatial parameters of objects. As a result, the proposed model has the potential for further improvement, meeting the requirements of Industry 5.0 and supporting the development of safe, reliable robotic systems focused on collaborative work with humans.

References

1. Samoilenko, H., & et al. (2024). Review for Collective Problem-Solving by a Group of Robots. Journal of Universal Science Research, 2(6), 7-16.
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., Yevsieiev, V., Nevliudov, I., & Uluhan, N. (2024). Constructing an Optimal Route for a Mobile Robot Using a Wave Algorithm. Journal of Natural Sciences and Technologies, 3(1), 282-289.
4. 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.
5. Basiuk, V., & et al. (2024). Command System for Movement Control Development. Multidisciplinary Journal of Science and Technology, 4(6), 248-255.
6. Yevsieiev, V., & et al. (2024). The Sobel algorithm implementation for detection an object contour in the mobile robot's workspace in real time. Technical Science Research in Uzbekistan, 2(3), 23-33.
7. Maksymova, S., & et al. (2024). The Lucas-Kanade method implementation for estimating the objects movement in the mobile robot's workspace. Journal of Universal Science Research, 2(3), 187-197.
8. 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.
9. 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.
10. 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.
11. 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.

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. 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.
14. 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.
15. Sotnik, S., Matarneh, R., & Lyashenko, V. (2017). System model tooling for injection molding. International Journal of Mechanical Engineering and Technology, 8(9), 378-390.
16. 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.
17. Гиренко, А. В., Ляшенко, В. В., Машталир, В. П., & Путятин, Е. П. (1996). Методы корреляционного обнаружения объектов. Харьков: АО “БизнесИнформ, 112.
18. 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.
19. 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.
20. 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.
21. 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.
22. 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.
23. 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.

24.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.
25.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.
26.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.
27.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.
28.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.
29.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.
30.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.
31.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.
32.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.
33.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.

34. 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.
35. 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.
36. 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.
37. 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.
38. 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.
39. 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.
40. 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.
41. 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.
42. Rahman, M. M., & et al. (2024). Cobotics: The Evolving Roles and Prospects of Next-Generation Collaborative Robots in Industry 5.0. Journal of Robotics, 2024(1), 2918089.
43. Zafar, M. H., & et al. (2024). Exploring the synergies between collaborative robotics, digital twins, augmentation, and industry 5.0 for smart manufacturing: A state-of-the-art review. Robotics and Computer-Integrated Manufacturing, 89, 102769.

ISSN: 3060-4745 IF(Impact Factor)10.41 / 2024
Volume 2, Issue 1
44. Panagou, S., & et al. (2024). A scoping review of human robot interaction research towards Industry 5.0 human-centric workplaces. International Journal of Production Research, 62(3), 974-990.
45. Doyle Kent, M., & Kopacek, P. (2021). Do we need synchronization of the human and robotics to make industry 5.0 a success story?. In Digital Conversion on the Way to Industry 4.0: Selected Papers from ISPR2020, September 24-26, 2020 Online-Turkey, Springer International Publishing. 302-311.
46. Prassida, G. F., & Asfari, U. (2022). A conceptual model for the acceptance of collaborative robots in industry 5.0. Procedia Computer Science, 197, 61-67.
47. Coronado, E., & et al. (2022). Evaluating quality in human-robot interaction: A systematic search and classification of performance and human-centered factors, measures and metrics towards an industry 5.0. Journal of Manufacturing Systems, 63, 392-410.