Vision-based Robot System for Object Manipulation

Abstract. The paper presents a robotic system for object manipulation based on information obtained from a camera. The developed system enables the differentiation of four classes of objects with regular geometric shapes. To achieve that, a semantic segmentation model was trained using a set of images of objects in different positions. An algorithm for objects' position and orientation determination was developed so objects can be placed in arbitrary positions and orientations within the camera field of view. The developed algorithms ensure the necessary information for automatic robot programming for moving the objects to desired poses. To prove the proposed concept on the 4-axis SCARA robot equipped with a vacuum gripper for object grasping, a camera calibration procedure was performed and necessary coordinate transformations were obtained. The verification of the developed system was conducted through several experiments. The experiments showed good reliability of the trained model for objects' classification and accurate positioning of the robot end-effector above the objects.

Keywords: SCARA robot, Vision System, Object Manipulation, Convolutional Neural Network

1 Introduction

Industrial robots today have a key role in manufacturing systems through tasks such as material and part manipulation, assembly, inspection, machining, additive manufacturing, etc. The need for extremely flexible manufacturing systems, which are a consequence of increasingly personalized production, imposes requirements on robot manufacturers to improve the functions, technical characteristics, as well as control and programming systems of industrial robots. Today, robots are capable of making decisions learning from experience in different situations, and adapting to dynamic changes in the environment while performing tasks [1].

Visual sensors integrated with the robotic arms increase the robustness of the robotic system and help the robot in better sensing its surroundings [2]. The vision system also makes the robots capable of performing optical inspections, sorting objects and taking measurements. Using vision system, the robot can determine where the objects are located in its workspace and perform assigned tasks.

The need for object recognition systems is met in multiple industrial applications, where different objects of variable shapes and sizes should be handled [3]. To achieve this, images need to be processed and analyzed through advanced algorithms, including deep learning. These algorithms could be divided into three classes: 1) image classification, which determines the presence of specific objects in image data, 2) object detection, which identifies instances of semantic objects within predefined categories, and 3) image segmentation, which breaks down images into distinct segments for analysis [4]. Based on the previously mentioned it can be concluded that robots are blind machines that move according to their programming without robotic vision and artificial intelligence [5].

The selected robot for this research is the SCARA (*Selective Compliance Assembly Robot Arm*) robot. As highlighted in [6], Professor Hiroshi Makino from Yamanashi University in Japan invented and developed the SCARA robot together with his colleagues and industry partners. The SCARA robot is an industrial robot widely used in material handling and assembly tasks. The prototype of the 4-axis SCARA robot, presented in [7, 8] was used for testing the proposed concept of a vision-based system for object manipulation.

2 Outline of the concept

The paper presents the developed system for manipulating different objects using the 4-axis SCARA robot and a developed vision system equipped with a semantic segmentation model. Fig. 1 shows the developed system including software and hardware integration with the realized prototype of the 4-axis SCARA robot and its control system. This integration allows the robot to be programmed automatically based on the information obtained from the camera. To prove the presented concept, a convolutional neural network was trained to differentiate 4 classes of objects with regular geometric shapes (triangle, rectangle, square, and circle). These objects can be placed in the camera field of view in an arbitrary position and orientation.

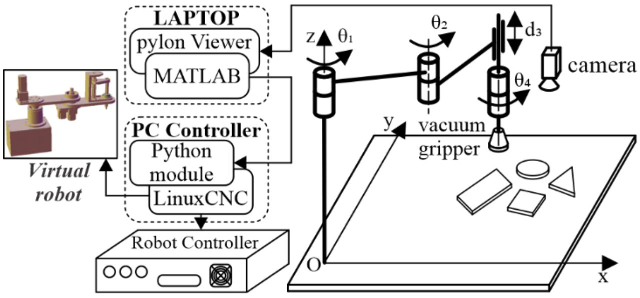

Fig. 1. Outline of the concept

The developed open architecture control system of the SCARA robot based on *LinuxCNC* allowed camera implementation and automatic programming of the robot for manipulating objects. As shown in Fig. 1, the control system consists of two PCs and a robot controller. The designed robot controller contains all the necessary electronic components for motion control of the robot with stepper motors and vacuum gripper operation.

An external laptop with *pylonViewer* installed allows image acquisition. The captured images were processed with a developed algorithm in *MATLAB* to differentiate objects in the scene and estimate their position and orientation. The estimated coordinates of the objects are sent to the developed *Python module* within the PC controller.

The task of the developed *Python module* is to generate robot commands, i.e. program, based on the received information from the *MATLAB* program and prepare the robot for automatic program execution. Given that the open architecture control system supports programming the robots in G-code, it was necessary to adapt some M functions for vacuum gripper operation. The developed and implemented virtual robot enabled testing of the developed system before putting the real robot into operation. When all conditions for starting the robot are met, the objects are moved from their current location to the pre-programmed position and orientation.

3 Object pose estimation

The process of determining the position and orientation of objects in the image relative to the robot's coordinate system consists of three parts. The first part includes dividing the image into significant regions (regions of interest) and assigning class labels to each pixel in the image using semantic segmentation. The second part involves determining the centroid and orientation of the regions of interest in the input image and assigning those values to the appropriate classes of objects. The third part involves camera calibration, which ensures the transformation of the object’s position and orientation from the image coordinate system to the robot’s coordinate system.

3.1 Semantic segmentation

Semantic segmentation is a computer vision task that assigns a class label to pixels using a deep learning algorithm [9, 10]. This technique identifies collections of pixels and classifies them according to various characteristics. As a result, a segmentation map of an input image is created. This map represents a reconstruction of the original image in which each pixel has been color-coded by its semantic class. To do so, the semantic segmentation model proposed in this paper uses a convolutional neural network that is trained to identify and differentiate objects in the image.

To train the network, images of objects in different positions are taken. The collected original images are scaled to an image size of 54x96 pixels and converted to grayscale with depth information of one byte for each pixel. Using the *Image Labeler MATLAB* app pixels of these images are interactively labelled and exported. Five classes are labeled on the images including pixels referring to background, square, triangle, circle, and rectangle. This collection of images and its corresponding collection of pixel labeled images are expanded by applying image and pixel label augmentation to training data.

With the training dataset prepared in this way and by finding adequate parameters, a network was trained for identifying objects in an image. The trained network is a convolutional neural network with an encoder/decoder depth of 2, 4 convolution layers, and with filter size of [7, 9]. With the set of 108 training images, evaluated metrics of the trained semantic segmentation network showed global accuracy of 0.78.

As mentioned earlier, output from the network is the segmentation map of an input image which consists of colored pixels assigned to its classes. Fig. 2a shows an example of an input image of objects in the robot's workspace that is scaled and converted to grayscale, while Fig. 2b shows the result from the trained network (segmentation map).

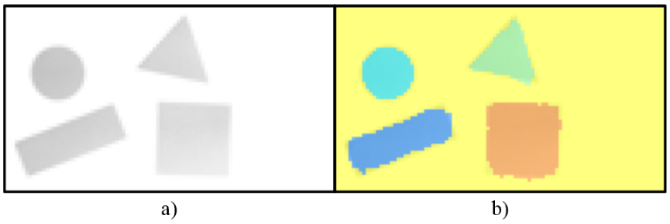

Fig. 2. Input image and the corresponding output generated by the trained network

3.2 Objects' position and orientation

Based on the pixels grouped in the classes and the corresponding output from the network (Fig.2b), centroids are determined for objects belonging to non-background classes. The values of the centroid coordinates obtained in this way are up-scaled 20 times so that these values correspond to the values of the coordinates on the original image from the camera with an image size of 1080x1920 pixels.

The reduced resolution of the images used as input to the network decreases the accuracy of classifying edge pixels. There is uncertainty about whether an edge pixel belongs to the object or the background. This results in an incorrect and irregular shape of the object in the image and increases the error of determining the value of the centroid coordinates.

To reduce this error, the centroid's coordinates of the objects were determined on the original image with the not-scaled resolution. The centroid's affiliation to the corresponding object in the original image is determined based on the Euclidean norm of the vector $\Delta\vec{r}_{i,j}$ which represents the difference between the centroid's position vector of the object in the segmented image $\vec{r}_i$ and the centroid's position vector in the original image $\vec{r}_j$. For each object in the original image, the difference between the position vector $\Delta\vec{r}_{i,j}$ of its centroid $\vec{r}_i$ and the position vector of the centroid of all objects in the segmented image $\vec{r}_j$ is determined. Based on the minimum value of the Euclidean norm of the vector $\Delta\vec{r}_{i,j}$, the affiliation of the centroids to each object in the original image was determined. Fig. 3 illustrates the previously described procedure on the example of determining the centroid for a square-shaped object in image. The orientation of the objects in the image was determined using the *MATLAB* function *regionprops*, which determines the orientation of the objects as the angle between the x-axis and the major axis of the object ellipse in the image.

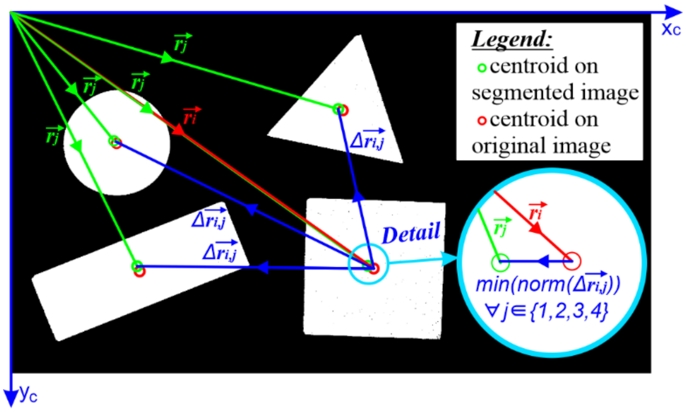
xc
yc
Legend:
○ centroid on segmented image
○ centroid on original image
$\vec{rj}$
$\vec{ri}$
$\vec{\Delta ri,j}$
Detail
min(norm($\vec{\Delta ri,j}$))
∀j∈{1,2,3,4}

Fig. 3. Determination of the objects' centroids in original image

3.3 Camera calibration

The main goal of the camera calibration process is to find the internal and external camera parameters. The matrix K_{3x3} represents the matrix of the internal camera parameters. The rotation matrix R_{3x3} and the translation vector $\boldsymbol{t}_{3x1}$ represent the external camera parameters. Based on these camera parameters, the transformation matrix $P = A \cdot [R \;\; \boldsymbol{t}]$ is obtained and it maps a point from the 3D world to the 2D image plane. Based on the pinhole camera model [11], the transformation that determines the x and y coordinates in the selected reference coordinate system of a pixel from the 2D image plane is expressed as:

$$s \cdot {}^{w}\boldsymbol{p} = H^{-1} \cdot {}^{p}\boldsymbol{p} \tag{1}$$

The position vector of the point P_w relative to the reference coordinate system, denoted by ${}^{w}\boldsymbol{p}$, is defined as:

$${}^{w}\boldsymbol{p} = [X_w \;\; Y_w \;\; 1]^T \tag{2}$$

The position vector of the pixel p relative to the image coordinate system, denoted by ${}^{p}\boldsymbol{p}$, is a vector of homogeneous coordinates given as:

$${}^{p}\boldsymbol{p} = [u \;\; v \;\; 1]^T \tag{3}$$

The homography matrix [12], denoted by H, is determined by removing the third column of the previously mentioned matrix P:

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \tag{4}$$

The calibration procedure consisted of taking several pictures of a chessboard of known dimensions in different positions and orientations within the camera field of view. During the camera calibration process, image distortion coefficients are also determined to eliminate it because equation (1) is only valid for images without distortion. The camera calibration was performed using *Camera Calibrator MATLAB* application. During camera calibration, the calibration template is placed in the position and orientation that match the known selected reference coordinate system pose in the robot's workspace. Fig. 4a shows relative position between the selected reference coordinate system ($O_w x_w y_w z_w$) and the coordinate system of the robot ($O_0 x_0 y_0 z_0$).

The coordinates from the selected reference coordinate system are transformed to the robot’s coordinate system, assuming relative positions between coordinate systems shown in Fig. 4a, using equation:

$$ {}^{\mathbf{0}}\boldsymbol{p} = [X_0 \;\; Y_0 \;\; 0 \;\; 1]^T = \boldsymbol{T} \cdot {}^{w}\boldsymbol{p} = \begin{bmatrix} 0 & 1 & 0 & 350 \\ 1 & 0 & 0 & -40 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot [X_w \;\; Y_w \;\; 0 \;\; 1]^T \tag{5} $$

By transforming the centroid coordinates of the objects into the robot's coordinate system, the position of the objects is obtained. The orientation of the objects is the same as in the image coordinate system due to the parallelism of the coordinate systems, only of the opposite sign. With obtained position and orientation of the objects relative to the robot’s coordinate system, it is possible to generate a program to perform the task of manipulating these objects.

4 Experiment

The verification of the developed system for manipulating objects with a 4-axis SCARA robot based on information from the vision system and neural networks was performed by several experiments. The goal of the experiment was to recognize different objects placed within the robot's workspace, allowing the robot to pick the desired object and place it in a specified position and orientation. The chosen objects for manipulation have regular geometric shapes (square, triangle, rectangle, and circle) and for their recognition, the semantic segmentation model was created.

The task is as follows: in the workspace of the robot, place the objects in an arbitrary position and orientation, and based on the vision system, perform automatic programming and control of the robot for placing each of them in the predetermined position and orientation.

The preparation procedure for the verification consists of several steps. Firstly, it is necessary to calibrate the camera to obtain the parameters for performing the previously mentioned transformations. To execute a manipulation task PC controller has to be started, objects placed in the robot workspace, and adequate lighting has to be ensured. Also, it is necessary to physically determine the *z* coordinate for picking the objects by vacuum gripper.

After the objects were placed in the robot’s workspace, camera took the image of the workspace. The image was processed using the developed algorithm and all the necessary information for automatic programming were extracted. Before the automatic program generation, the developed *MATLAB* program generates the check image with labeled recognized objects and their position and orientation. After confirmation, the program was automatically executed and objects are moved as shown in Fig. 4b.

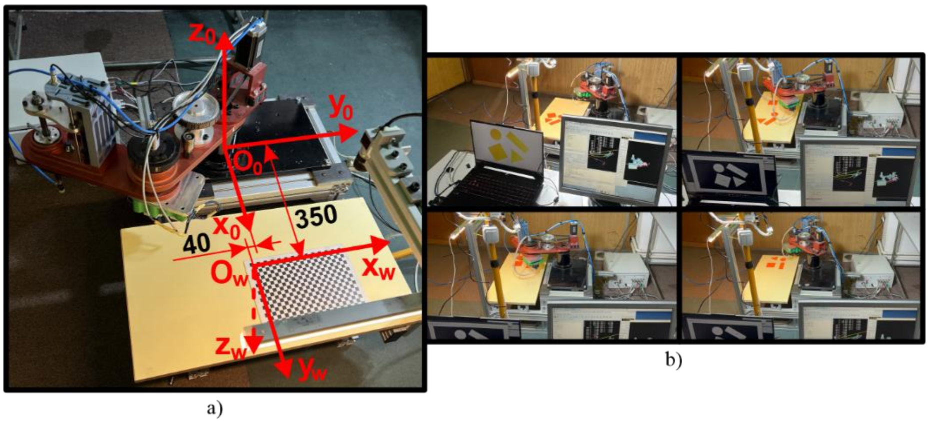

Fig. 4. Relative position of the coordinate systems (a); Manipulating with objects (b)

Based on the several experiments it can be concluded that the system was successfully developed. Also, it is noticed that the end-effector of the robot positions precisely above the physical objects centroid. Orientation of the objects after placing them in the specified pose is programmed to be parallel with the robot’s *y*-axis, and physically objects are placed with good accuracy in that orientation.

5 Conclusion

Applications of industrial robots are rapidly expanding with the constant improvement of their functions, technical characteristics, as well as control and programming systems. The presented research includes the development of the robotic system for manipulating different objects using the 4-axis SCARA robot and the developed vision system equipped with a semantic segmentation model implemented in the developed open architecture control system. The robot’s control system includes a virtual robot that represents a digital shadow of the presented SCARA robot and allowed all the developed control algorithm testing before putting the real robot into operation.

To prove the developed system, the semantic segmentation model was trained to differentiate objects with regular geometric shapes. Still, the proposed concept can be easily reconfigured to work in a real-world application. The trained model in the developed control algorithm showed good reliability through the experiments.

The conducted experiments showed accurate positioning of the robot end-effector above the objects pointing out two facts that are interconnected. The first fact is that the developed image processing algorithm is accurate enough for the described task. The second fact is that the camera calibration procedure is well-done and all the necessary transformations are correct.

Future research will include development of the vision-based robotic system equipped with semantic segmentation of the point clouds obtained from the RGB-D camera for performing bin picking task.

6 Acknowledgment

This research was supported by the Science Fund of the Republic of Serbia, grant No. 17801, Cybersecurity of Motion Control Systems in Industry 4.0 - MCSecurity as well as by the Ministry of Science, Technological Development and Innovations of the Serbian Government under the contract No. 451-03-65/2024-03/200105.

References

1. Javaid M., Haleem A., Singh R.P. and Suman R.: Substantial capabilities of robotics in enhancing industry 4.0 implementation. Cognitive Robotics 1, 58-75 (2021).
2. Phuong L. H., Cong V. D. aand Hiep T. T.: Design a Low-cost Delta Robot Arm for Pick and Place Applications Based on Computer Vision. FME Transactions 51, 99-108 (2023).
3. Tsarouchi P., Matthaiakis S., Michalos G., Makris S., Chryssolouris G.: A method for detection of randomly placed objects for robotic handling. CIRP Journal of Manufacturing Science and Technology 14, 20-27 (2016).
4. Manakitsa, N.; Maraslidis, G.S.; Moysis, L.; Fragulis, G.F.: A Review of Machine Learning and Deep Learning for Object Detection, Semantic Segmentation, and Human Action Recognition in Machine and Robotic Vision. Technologies 12(2), 15 (2024).
5. Robot Vision System: What You Need to Know, https://www.tm-robot.com/en/robot-vision-system/, last accessed 2025/01/20.
6. Makino H.: Development of the SCARA. Journal of Robotics and Mechatronics 26(1), 5-8 (2014).
7. Miljkovic Z., Slavkovic N., Momcilovic B., Milicevic D.: Development of a Domestic 4-axis SCARA Robot. In: Proceedings of the XI International Conference Heavy Machinery, pp. P1-P9, University of Kragujevac, Faculty of Mechanical Civil Engineering, Kraljevo (2023).
8. Momčilović, B., Slavković, N., Vorkapić, N., Živanović, S.: Prototype of Scara-Type Industrial Robot. In: Proceedings of the 44. Jupiter Conference, pp. 3.30-3.37, University of Belgrade – Faculty of Mechanical Engineering, Belgrade (2024).
9. Guo, Y., Liu, Y., Georgiou, T. et al.: A review of semantic segmentation using deep neural networks. Int J Multimed Info Retr 7, 87–93 (2018).
10. Semantic Segmentation Using Deep Learning, https://www.mathworks.com/help/vision/ug/semantic-segmentation-using-deep-learning.html, last accessed 2025/01/20.
11. Open Source Computer Vision - OpenCV, Camera Calibration and 3D Reconstruction, https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html, last accessed 2025/01/20.
12. Zhang Z.: A flexible new technique for camera calibration. IEEE Transactions on pattern analysis and machine intelligence 22(11), 1330-1334 (2000).