

# Vision-based control of small educational parallel selective compliance assembly robot arm robot

Michał Batsch<sup>1\*</sup> , Dariusz Nowak<sup>1</sup>, Dawid Wydrzyński<sup>1</sup>, Łukasz Kochmański<sup>1</sup>

<sup>1</sup> Faculty of Mechanical Engineering and Aeronautics, Rzeszów University of Technology, Al. Powstańców Warszawy 8, Rzeszów, Poland

\* Corresponding author's e-mail: mbatsch@prz.edu.pl

## ABSTRACT

Vision-based control in robotics offers versatile automation; however, accessible educational platforms for exploring its integration with AI are still limited. This paper addresses this gap by presenting a small, 3D-printed parallel SCARA robot designed specifically for educational purposes. We provide details on its construction and demonstrate its application in laboratory exercises, which cover inverse and forward kinematics, vision-based tip positioning, and object detection. Notably, we investigate both supervised (using convolutional neural networks) and unsupervised (through autoencoder latent space exploration) approaches for classifying faulty parts. The unsupervised method achieved high performance, with a precision of 1.00, recall of 0.96, and an F1-measure of 0.98, which is comparable to the supervised approach that yielded a precision of 0.98, recall of 0.97, and an F1-measure of 0.97. This work contributes to the development of a low-cost platform and demonstrates the effectiveness of unsupervised AI techniques for vision-based robotic fault detection in educational settings, paving the way for more advanced AI-integrated robotics curricula.

**Keywords:** SCARA robot, vision system, neural networks, image processing, classification, additive manufacturing.

## INTRODUCTION

The rapid advancement of robotics and artificial intelligence has significantly transformed modern industrial environments, particularly in the areas of logistics, manufacturing, and automated assembly. Autonomous and semi-autonomous robotic systems are increasingly deployed in scenarios that demand precision, efficiency, and adaptability—ranging from warehouse automation to therapeutic applications and human-robot collaboration in production lines. A central challenge in these domains is the development of effective planning and control strategies for multi-robot systems operating under constraints such as limited communication, dynamic environments, or human presence. Traditional approaches to autonomous planning often rely on centralized systems or explicit inter-robot communication. However, recent studies, such as Grzejszczak and Nocon [1], demonstrate that decentralized and cooperative

planning methods, even under non-communicative conditions, can substantially enhance performance in tasks like parcel transportation in warehouses. Their simulation-based work shows that cooperation, in the form of intelligent task allocation and physical interaction strategies (e.g., package hand-offs), can significantly reduce idle time and improve task efficiency. Meanwhile, the integration of mobile and collaborative robots is reshaping the landscape of industrial automation. Boschetti and Minto [2] explore various control strategies for mobile collaborative robots, focusing on admittance controllers to facilitate safe and intuitive human interaction. Complementing this, Gusan et al. [3] provide a comparative perspective on the roles of industrial versus collaborative robots, emphasizing their functional benefits and relevance in non-conventional and intelligent manufacturing systems. In parallel, touch-based human-robot interaction is emerging as a key factor in therapeutic and assistive robotics. Mitsuoka et al. [4] propose

a semi-autonomous touch system that blends user input with robot-initiated contact, aiming to reduce discomfort and enhance emotional engagement in users – demonstrating the growing interdisciplinary nature of robotics research.

The advancement of mobile and collaborative robotics has become central to addressing modern industrial, logistic, and social challenges. Burghardt et al. [5] modeled the dynamics of cooperating wheeled mobile robots transporting large objects, using Lagrange equations and the projective method to obtain a control-oriented system free of unknown friction forces. Complementary work by Buratowski et al. [6] and Szuster et al. [7] explored hierarchical and fuzzy-neural control strategies for mobile robot formations and autonomous navigation, respectively. Lee et al. [8] and Andersson et al. [9] investigated collaborative robots in industrial environments, emphasizing the importance of safety, usability, and control accuracy for effective human-robot interaction. Ferraguti et al. [10] proposed a standardized methodology for benchmarking collaborative robots, providing insight into their capabilities within the context of Industry 4.0. In the domestic and service sector, Yamamoto et al. [11] evaluated user affinity with autonomous and remote-controlled robots in fetch-and-carry tasks, suggesting hybrid control modes as a practical solution. Research by Rykala et al. [12] demonstrated the use of Ultra Wideband (UWB) technology for tag tracking with mecanum-wheeled robots, while Xu et al. [13] presented a neural network-based model identification approach for autonomous robot kinematics. Moreover, Burghardt et al. [14–17] have contributed significantly to mechatronic design and practical implementation, showcasing applications such as virtual reality-based robot programming, mobile robot prototyping, and robot-assisted inspection of turbine geometries. Finally, Daneshjo et al. [18] examined vibrodiagnostics in robotic maintenance, presenting vibration analysis as a tool for assessing the health of industrial robots.

SCARA (selective compliance assembly robot arm) robots have been the subject of extensive research due to their versatility in manufacturing and educational contexts. These robots are often explored for their unique kinematic configurations and potential applications in industrial and miniature environments. Van Helvoort et al. [19] presented a low-order linear fractional transformation (LFT) model for a double SCARA robot,

demonstrating its effectiveness in gain-scheduling control systems. Their method utilized analytical modelling and frequency response measurements to validate the system's stability and applicability in industrial settings. This work underscores the potential of SCARA robots in precision-driven environments through advanced control design. Similarly, Coman et al. [20] focused on a double-arm SCARA robot with a five-link parallel mechanism. Their study covered physical structure development, kinematic modeling, and integration of control systems to perform pick-and-place tasks. Their findings emphasize the importance of kinematic accuracy and repeatability in industrial scenarios. Siltala et al. [21] introduced the H-SCARA mini robot, which combines dual parallel kinematic structures. This compact design aligns with the modularity of desktop and micro-factory concepts, enabling efficient manufacturing with minimal workspace requirements. Their work highlighted the potential for reconfigurable assembly systems in constrained environments. Vuola et al. [22] discussed the broader scope of miniaturized production systems and their alignment with sustainable manufacturing trends. The presented flexible screwing cell exemplifies how macro-world assembly systems can be adapted for miniaturized applications, demonstrating the scalability of SCARA-based solutions. Salzmann et al. [23] proposed a miniature robot for otologic surgery, featuring a five-degree-of-freedom parallel structure tailored for precise bone drilling. This study exemplifies how parallel kinematic configurations like those of SCARA robots can meet the stringent accuracy demands of medical applications.

The integration of vision systems into robotic applications has been extensively explored to enhance autonomy and adaptability. Vision-based control techniques are pivotal for tasks requiring precision and adaptability in dynamic environments. Ali et al. [24] demonstrated the integration of a vision system with an existing Scorbot manipulator for object sorting in industrial applications. They addressed challenges such as communication protocols, camera-robot coordination, and system integration, paving the way for adaptable manufacturing solutions. Mehta and Burks [25] developed a vision-based control system for robotic fruit harvesting. Their cooperative visual servo controller utilized fixed and camera-in-hand systems to ensure precise targeting and harvesting. The controller's stability analysis and practical evaluations highlighted the role of vision systems

in enhancing robotic efficiency in agricultural domains. Stieber et al. [26] investigated vision-based sensing and control for space manipulators, addressing challenges like structural flexibility and non-collocated sensor-actuator configurations. Their work demonstrated how photogrammetric image processing could achieve precise real-time positioning for payload handling in space. Wells et al. [27] utilized feedforward neural networks for vision-based robot positioning. By learning the implicit relationship between pose displacements and global image descriptors, their method achieved high accuracy in industrial inspection tasks, showcasing the potential of AI (artificial intelligence) in enhancing visual servoing. Corke and Hager [28] provided an extensive overview of vision-based control methodologies, emphasizing the interplay between vision and control systems. They discussed key issues like dynamic performance, feature extraction, and future research trends. A comprehensive review by Hashimoto [29] further categorized vision-based control approaches, focusing on robust and globally stable controllers. Their work highlights the evolution of vision-based servo mechanisms and their critical role in advancing robotic manipulators. The work of Bochen and Ambrozkiwicz [30] investigates how light intensity impacts the performance of vision systems used in collaborative robots like ABB Yumi, offering insights into the environmental dependencies of visual object detection.

The inclusion of vision-based control in the education of future robotics and mechatronics engineers is paramount to prepare them for the challenges of modern industry and research. Vision systems are integral to advanced robotics applications, enabling robots to operate in unstructured, dynamic environments with enhanced perception and adaptability. By incorporating vision-based control into educational programs, students gain hands-on experience in critical areas such as image processing, computer vision, and visual servoing. These skills are essential for developing solutions to real-world problems, ranging from autonomous navigation and object manipulation to defect detection and quality control in manufacturing. Furthermore, exposure to the integration of vision systems with robotic platforms fosters an interdisciplinary approach, combining principles of electronics, programming, and mechanical design. This holistic education equips future engineers to innovate in rapidly evolving fields such as autonomous systems, collaborative

robots, and AI-powered robotics, ensuring their competitiveness and adaptability in the workforce. This project aimed to develop an easy-to-build and deploy educational robot, which may be used during classes to study machine vision-based control and classification. The remainder of this paper is organized as follows:

- Section 2 describes the design of the educational parallel SCARA robot, detailing its geometry, kinematics, control system configuration, and manufacturing process.
- Section 3 focuses on the sample laboratory exercises developed for the robot, starting with a pick-and-place task that includes the general control algorithm, synchronous movement of stepper motors, object detection, and tool center point positioning. The section then delves into the supervised classification of parts, covering data preparation and the use of deep convolutional neural networks. It concludes with unsupervised classification of parts, exploring data preparation, convolutional autoencoders, and latent space exploration. Here we present the results and discussion of the experiments conducted during the laboratory exercises.
- Section 4 provides a summary of the findings and conclusions drawn from the study.

## ROBOT DESIGN

This section presents the design of a double SCARA manipulator, covering its geometry, kinematics, manufacturing processes, and the control system used.

### Geometry and kinematics

The robot was designed as a parallel SCARA manipulator driven by two stepper motors controlled by Raspberry Pi Zero 2W. The 3D CAD model of the designed manipulator is shown in Figure 1.

As the gripper, the electromagnet is used, which may manipulate objects into one from selected boxes attached, the camera observes the working area of the manipulator. The kinematic diagram and top view of the manipulator are presented in Figure 2. Two stepper motors are directly connected (by joints  $O_L$  and  $O_R$ ) to the first pair of arms  $|A_L O_L|$ ,  $|A_R O_R|$  (length  $l_1 = 75$  mm) which are connected by revolute joints (points  $A_L$  and  $A_R$ ) with the second pair of arms  $|A_L TCP|$ ,  $|A_R TCP|$  (length  $l_2 = 95$  mm). The angles of rotation of

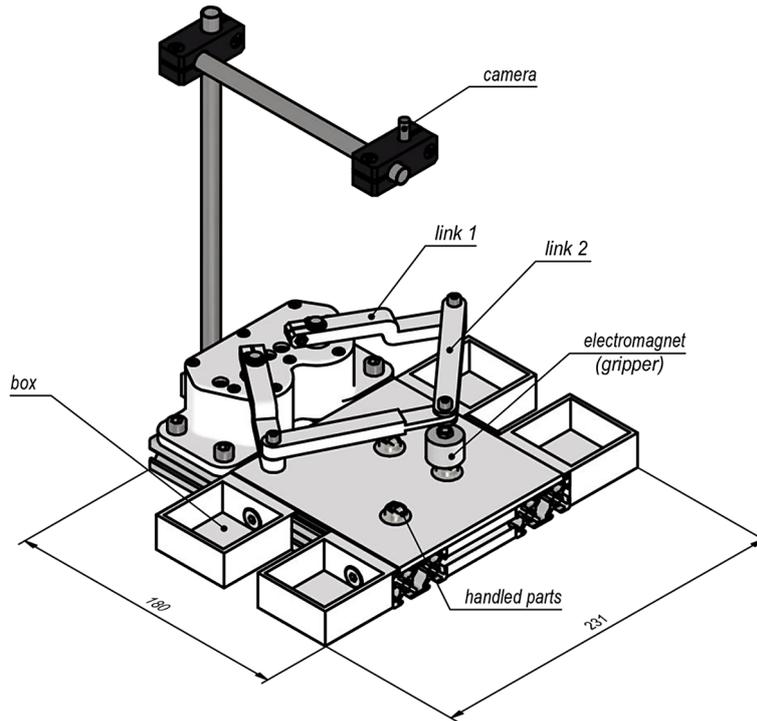


Figure 1. Isometric view of the parallel SCARA manipulator

the first pair of arms  $\varphi_L$  and  $\varphi_R$  define the position of the centre point of the tool (TCP). The points  $O_L$  and  $O_R$  are shifted through the horizontal coordinate system axis  $x$  against the vertical axis  $y$  by  $e = 22.5$  mm.

Based on Figure 2 we may find the inverse kinematics of the manipulator given by Equation 1:

$$\begin{aligned} \varphi_R &= \text{atan2}(y_{TCP}, x_{TCP} - e) - \\ &\quad - \arccos \frac{(x_{TCP} - e)^2 + y_{TCP}^2 + l_1^2 - l_2^2}{2l_1 \sqrt{(x_{TCP} - e)^2 + y_{TCP}^2}} \\ \varphi_L &= \text{atan2}(y_{TCP}, x_{TCP} + e) + \\ &\quad + \arccos \frac{(x_{TCP} + e)^2 + y_{TCP}^2 + l_1^2 - l_2^2}{2l_1 \sqrt{(x_{TCP} + e)^2 + y_{TCP}^2}} \end{aligned} \quad (1)$$

where:  $\text{atan2}(y, x)$  is the 2-argument arctangent function,  $x_{TCP}$  and  $y_{TCP}$  are the tool centre point (TCP) coordinates in the coordinate system  $x$ - $y$ . Based on Figure 2 we can also derive the forward kinematics. By solving the system of Equations 2 for  $\gamma_L$  and  $\gamma_R$ :

$$\begin{cases} l_1 \sin \varphi_L + l_2 \sin \gamma_L = \\ = l_1 \sin \varphi_R + l_2 \sin \gamma_R \\ -e + l_1 \cos \varphi_L + l_2 \cos \gamma_L = \\ = e + l_1 \cos \varphi_R - l_2 \cos \gamma_R \end{cases} \quad (2)$$

and then substituting them into (3):

$$\begin{aligned} x_{TCP} &= -e + l_1 \cos \varphi_L + l_2 \cos \gamma_L = \\ &= e + l_1 \cos \varphi_R - l_2 \cos \gamma_R \\ y_{TCP} &= l_1 \sin \varphi_L + l_2 \sin \gamma_L = \\ &= l_1 \sin \varphi_R + l_2 \sin \gamma_R \end{aligned} \quad (3)$$

we arrive at the coordinates of the tool centre point (TCP). Taking into account geometric boundary conditions such as:

- avoiding collision of the second pair of arms ( $\pi - \varphi_L - \varphi_R \geq 0.51$  rad),
- avoiding collision of arm  $l_1$  with  $l_2$  ( $\gamma_R + \varphi_R \geq 0.37$  rad and  $\gamma_L + \pi - \varphi_L \geq 0.37$  rad),
- avoiding collision of the first pair of arms with the body ( $\varphi_L \in [0.62$  rad,  $3.87$  rad] and  $\varphi_R \in [-0.73$  rad,  $2.51$  rad]),

we may obtain the workspace of the manipulator (Figure 3).

The gripper can reach every box and most of the worktable area, making it possible to perform pick-and-place tasks.

To produce manipulator parts, we have used Prusa i3 MK3S 3D printer (Figure 4a). This printer makes the use of melted and extruded manufacturing (MEM) also known as fused filament fabrication (FFF) or fused deposition modelling (FDM). This method involves building the model layer by layer by pressing filament through the nozzle, which is heated to melt the material being

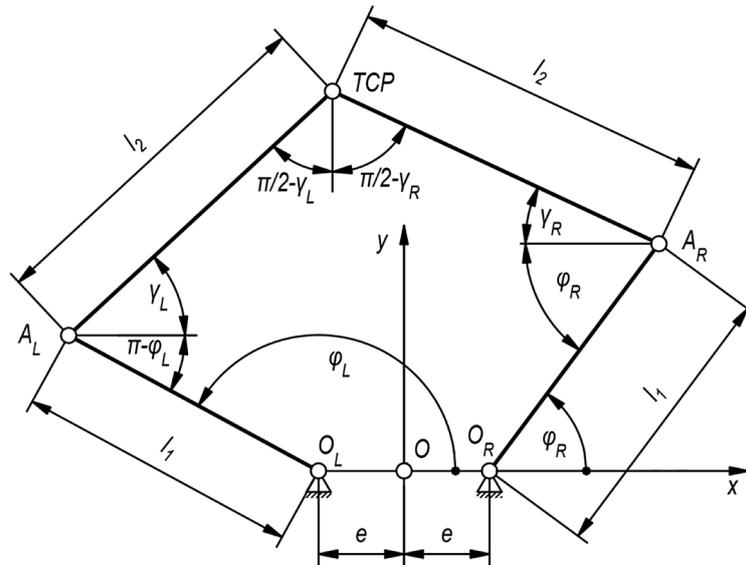


Figure 2. Kinematical structure of parallel SCARA manipulator

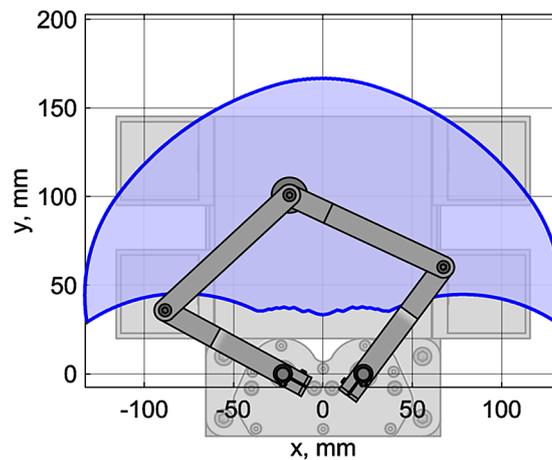


Figure 3. Workspace of the parallel SCARA manipulator

pressed. Parts of the manipulator were manufactured from ABS (acrylonitrile butadiene styrene) thermoplastic polymer, which is characterized by significant strength, resistance to shocks and relatively high melting temperature [31].

Models were arranged on the worktable of the 3D printer in the virtual environment (Figure 4b). Next the 3D printing parameters were chosen (Figure 4c). The basic parameters were 0.2 mm of layer thickness and 80% of parallel lines internal fill.

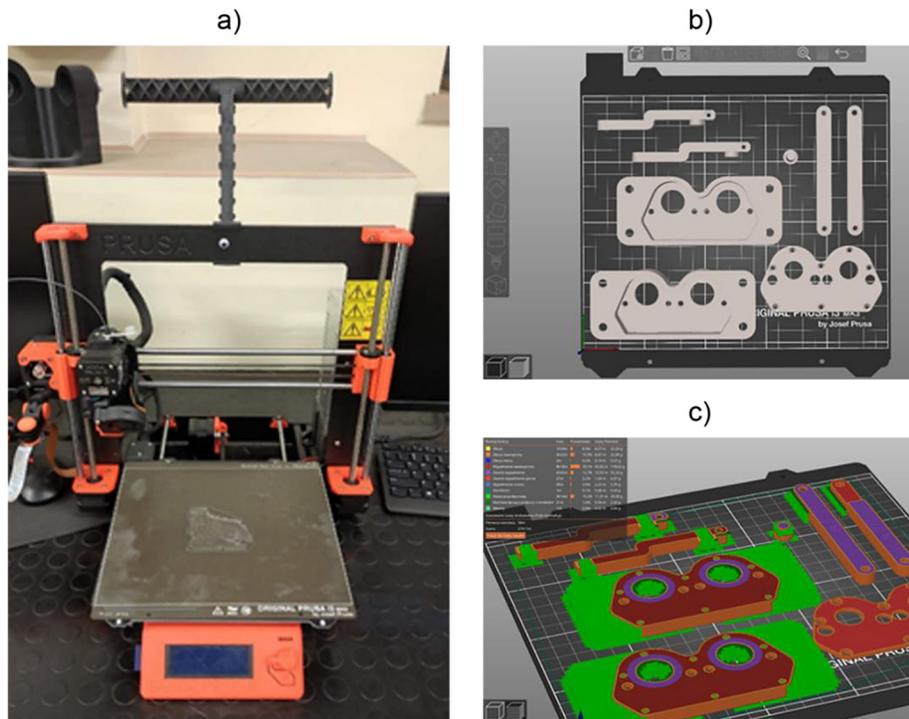
### Control system configuration

The control of the manipulator is realised by a Raspberry Pi Zero 2W microcontroller, which might be programmed with the aid of MATLAB

environment, Python, or C (Figure 5). The code can be deployed and tested remotely on the edge device via the WiFi network. The USB camera is connected to the first micro-USB port on the board. Control of the stepper motors (28BYJ-48) is performed with the aid of two stepper motor drivers (ULN2008) which are controlled by a Raspberry Pi board through the general purpose input/output interface (GPIO). This interface is also used for gripper control (electromagnet module SEN-MAG25N).

### SAMPLE LABORATORY EXERCISES

This section introduces sample laboratory exercises that can be performed during classes, namely:



**Figure 4.** Manipulator parts manufacturing: a) Prusa i3 MK3S 3D printer, b) parts arrangement on the worktable, c) view of the internal structure of models

pick and place task, supervised classification of parts, and unsupervised classification of parts.

### Pick-and-place task

Pick-and-place tasks are fundamental in robotics, involving the movement of an object from one location to another. These tasks are vital for automating processes in various industries, such as manufacturing, logistics, healthcare, and service robotics. This section introduces a general control algorithm for coordinating the synchronous movement of stepper motors, along with object detection and tool center point positioning.

#### General control algorithm

Here, we present the general control algorithm for the pick and place task. It can be realised by pseudocode given in Figure A.1 (Appendix). In this case, the code was implemented in MATLAB and deployed on the edge device with automatic C-code generation. To run the algorithm, we need several functions which will be explained further in subsequent section. The algorithm starts by defining the coordinates of the boxes (line 1) and getting the coordinates of the objects based on image processing (function `getObjCoords` – line 2). Then the number of

objects for handling is calculated (line 3). Moreover, the initial angular position of arms  $\varphi_{L0}$ ,  $\varphi_{R0}$  is calculated with the aid of inverse kinematics (function `invKin` – line 4) based on initial position of the TCP ( $x = 0$ ,  $y = 29.3$  mm). These initial values are then stored as current angular position as variables  $\varphi_L$ ,  $\varphi_R$  (lines 5 and 6). Next, the program iterates over all objects (loop from line 8). Within that loop, several consequential actions are performed. Firstly, the physical coordinates of centroids for each object are obtained (line 9) with the aid of function `px2xy` which transforms pixels into millimetres (refer to section “Tool centre point positioning”). Next, the angular position of arms  $\varphi_{Lr}$ ,  $\varphi_{Rr}$  is calculated that corresponds to the position of the centroid of the handled object (line 10). These angles are then used to calculate angles  $\Delta\varphi_L$ ,  $\Delta\varphi_R$  at which arms need to rotate in order to take the position (lines 18 and 19) corresponding to the absolute angles  $\varphi_{Lr}$ ,  $\varphi_{Rr}$ . Then stepper motors perform the desired movement (line 20) and the values of the current angles are updated (lines 21 and 22). Next the electromagnet is turned on (function `magOn` – line 16) and the procedure is repeated to move to the box defined by counter  $k$  (line 17). If TCP reaches the desired position the magnet is turned off (function `magOff` – line 23). At the end of the

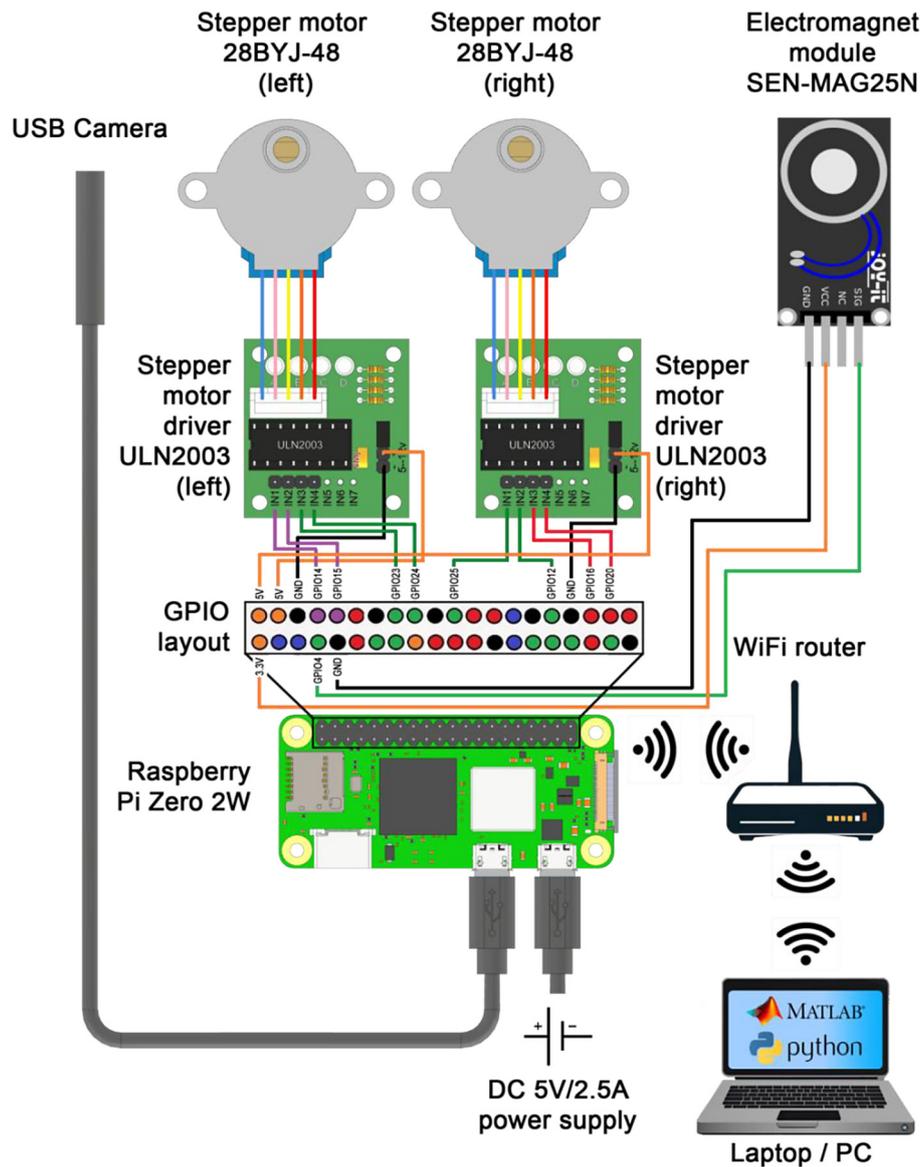


Figure 5. Control system configuration diagram

loop, condition is checked (line 25) in order not to exceed the number of boxes by counter  $k$ . This procedure is repeated for all objects detected. At the end of the program, TCP returns to initial position (lines from 29 to 32).

#### Synchronous movement of stepper motors

For synchronous rotation of stepper motors, the motorSync function is applied (Figure A.2 from an Appendix). Firstly in line 2 basic parameters of stepper motors are defined such as integrated gearbox reduction ratio ( $ratio = 64$ ), number of steps per revolution ( $st\_rev = 64$ ) and parameters of stepper motor control like number of steps ( $stepCount = 8$  for half-step operation) and step time ( $stepTime = 2$  ms). Next, the control

sequences are calculated for the left and right motors based on given angles of rotation (line 3). For that purpose, the impSeq function is used which is given by pseudocode from Figure A.3 (Appendix) and takes into account the sign of the angle. Further, the angles are converted from radians into the number of revolutions and next into number of steps (lines 4 and 5). Motors should achieve the desired angular displacement at the same time. There are two cases where the angular displacement of the left motor is greater than that of the right motor ( $stepsL \geq stepsR$ ) or when it is smaller ( $stepsL < stepsR$ ). The actual case is recognized by the conditional statement from line 9. In the first case ( $stepsL \geq stepsR$ ) dStep parameter is defined in line 7. It tells us how much faster

should left motor rotate in order to achieve the final position in similar time as right motor. In other words it gives us number of the steps of left motor which corresponds to one step of right motor. The above is realised in loops from line 9 and 11 in which setStepR and setStepL function are given by pseudocode from Figure A.4 (Appendix) and are responsible for setting the pin state for motors control. Similar procedure is undertaken for the second case ( $stepsL < stepsR$ ), where also dStep is defined in line 27. Here the right motor needs to keep up and move faster than the left one.

### Object detection

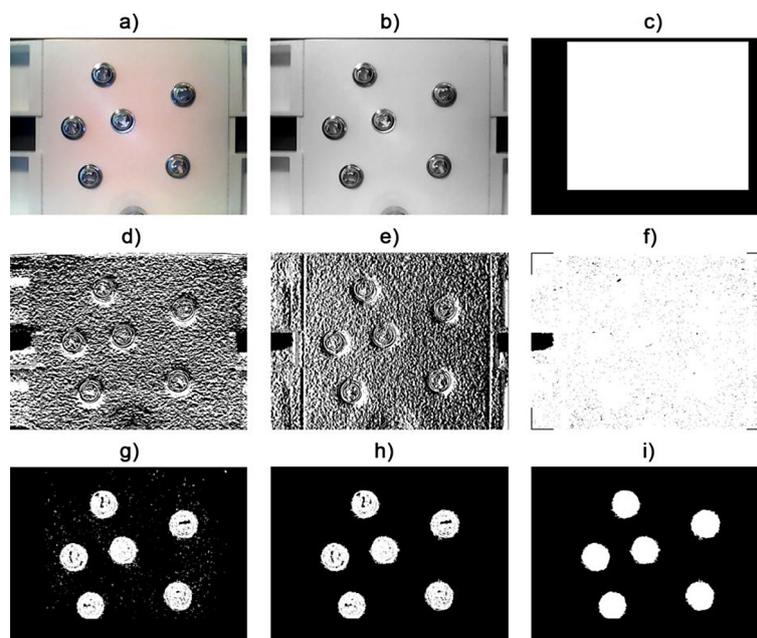
Object detection might be performed using the getObjCoord function (Figure A.5 from Appendix). This function returns coordinates of centroids of objects in pixels. The algorithm starts from capturing an image (line 2 – Figure 6a). Next, this image is transformed into a greyscale image (line 3 – Figure 6b). We also define the binary mask (M) within two loops (lines 5 and 6). The mask mentioned above (Figure 6c) corresponds to a working table area – only objects on this area are within our interest. Another step is to define a kernel for the convolution operation. In this study we use the Sobel kernel [32] (line 14). In addition, we produce convolved images H

and V, which correspond to horizontal and vertical edges (line 15 and 16 – Figure 6d,e). On line 17 the gradient-filtered image E (Figure 6f) is computed, where the symbol  $\odot$  denotes element-wise product. Next we produce binary image based on threshold value of 25, making use of logical indexing, which is then masked with previously defined binary mask M (line 18) - image  $B_w$ . Then the area opening is performed (line 19 – Figure 6g), so removing connected regions which have fewer pixels than the estimated object area. The hole filling operation is then performed (line 20 – Figure 6h). The final step of the algorithm is to define the hBlob object for the analysis and derivation of centroids (line 21 and 22). The subsequent steps of algorithm are visualised in Figure 6.

As a result, we obtain centroids coordinates, which are marked in Figure 7. The values in millimetres were calculated with the aid of the px2xy function described in Section 3.1.4.

### Tool centre point positioning

In order to get reasonable results, the machine vision system needs to be calibrated. Our main goal of calibration is to transform the pixel coordinates from the image into a system of coordinates of a manipulator with units of mm. To do that, we may use checkerboard pattern (Figure 8).



**Figure 6.** Image processing for object detection: a) original colour image, b) greyscale image, c) binary mask (M), d) convolved image – horizontal edges (H), e) convolved image – vertical edges (V), f) gradient filtered image (E), g) black and white image after thresholding and masking ( $B_w$ ), h) opened image ( $B_o$ ), i) image after hole filling ( $B_f$ )

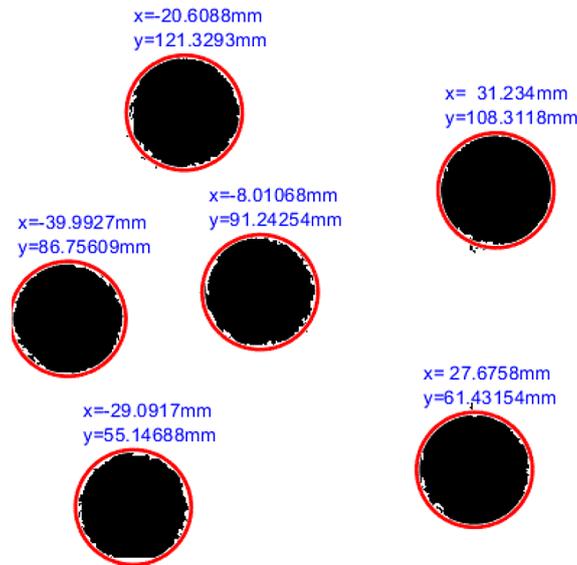


Figure 7. Results of the object detection algorithm

In this case, checkerboard pattern consists of 100 ( $10 \times 10$ ) black and white rectangles of width 10 mm. Two coordinate systems are introduced (Figure 8b):  $x$ - $y$  as an absolute coordinate system of the manipulator and  $s_c x_{im}$ - $s_c y_{im}$  which is a scaled coordinate system of an image. The scale coefficient  $s_c$  may be calculated by direct measurements of the checkerboard image. In this case, the scale was estimated as  $s_c = 0.241$  mm/px. From image measurements we may also estimate parameters  $v_x = s_c \cdot 340$  and  $v_y = s_c \cdot 212$  which gives an information about position of the image coordinate system in relation to the center of checkerboard. Furthermore, the checkerboard is placed in

such a way that  $t_y = 95$  mm. Finally, we arrive at transformation from image coordinate system to coordinate system of manipulator given as:

$$\begin{aligned} x &= s_c x_{im} - v_x \\ y &= -s_c y_{im} + v_y + t_y \end{aligned} \quad (4)$$

This is the simplest calibration method that does not take into account the skewness and distortion of an image resulting from camera lenses. In order to evaluate our calibration method, we may compare coordinates of points computed by Equation 4 and those obtained from an image (Figure 9).

The maximum eccentricity error between points estimated by Equation 4 and detected

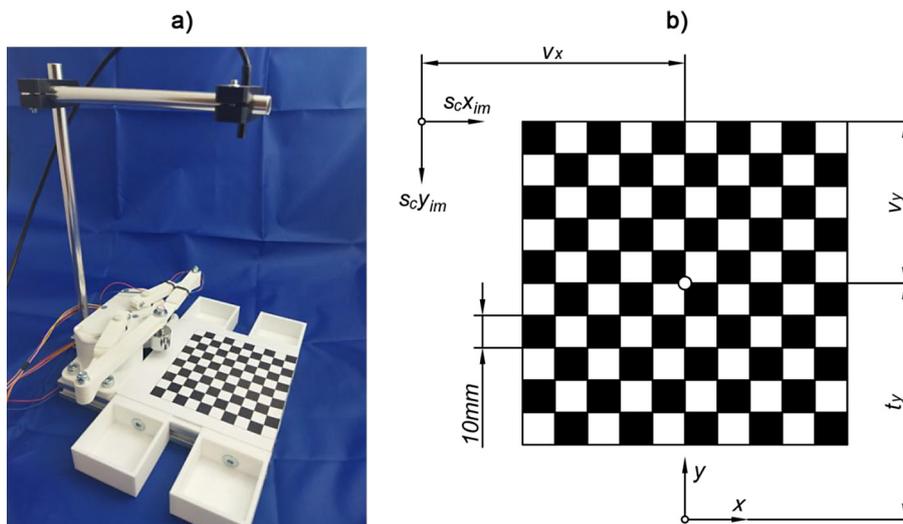


Figure 8. Calibration of vision system with the aid of checkerboard pattern: a) photograph, b) coordinate systems

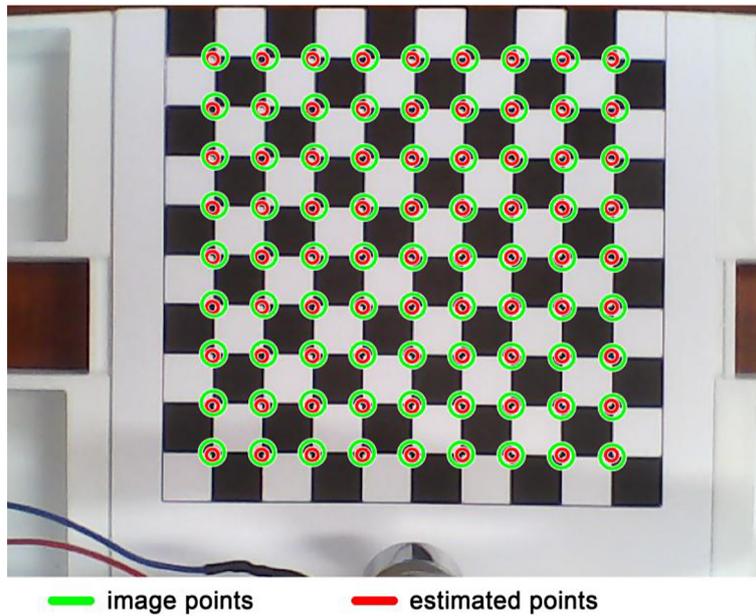


Figure 9. Evaluation of calibration

points is 0.9 mm thus for our application it was sufficient. It can be observed that an image is distorted and for a precise application, it would require a more sophisticated method of calibration such as the estimation of intrinsic and extrinsic camera parameters based on the camera model with lens distortion [33]. Such functionality is offered by MATLAB or OpenCV library and might be additionally used. All in all, the px2xy function is pretty straightforward (Fig. A.6 from an Appendix) and makes the use of calibration results and Equation 4. Tool centre point (TCP) is positioned in an open loop based on inverse kinematics given by Equation 1 which is introduced in the invKin function (Figure A.7 from Appendix). The invKin takes the desired TCP coordinates and converts them into absolute angles  $\varphi_R$  and  $\varphi_L$ . Such a method might be sufficient only for rough positioning like the one presented.

In order to compensate for backlash and assembly/manufacturing errors one may use closed-loop control with feedback from camera. The basic idea is that after taking the position based on inverse kinematics the image is captured, and then the TCP position is extracted to calculate the correction angles. This method is schematically illustrated in Figure 10.

Let us assume that it is required to achieve position TCP', however vision system recognizes tool centre point in position TCP. Based on the coordinates of the actual TCP and the desired position coordinates (TCP') we may obtain (with the

use of the invKin function) angles at which the arms need to rotate to take the desired position:

$$\begin{aligned} \Delta\varphi'_R &= -(\varphi'_R - \varphi_R) \\ \Delta\varphi'_L &= -(\varphi'_L - \varphi_L) \end{aligned} \quad (5)$$

To extract actual coordinates of tool centre point, we can use coloured marking, which was applied on screw head holding electromagnet (Figure 11a). Based on the histogram of an image of this marking, we can derive threshold values used in colour detection algorithm.

The colour detection algorithm (Figure A.8 from Appendix) aims to detect areas of pixels of the same colour (in our case - pink). The algorithm starts from acquiring an image for processing (line 1 – Figure 11a). Next we extract the red (R), green (G) and blue (B) bands (lines 2, 3 and 4). Further (lines 6–11) we define threshold values for each band, they were obtained based on histogram from Figure 11b). Then the maskThresh function (Figure A.9 from an Appendix) is used to generate red ( $M_R$ ), green ( $M_G$ ) and blue ( $M_B$ ) masks (lines 12, 13 and 14). The bands and their corresponding masks are presented in Figure 12.

The purple mask ( $M_p$ ) is calculated as the element-wise product of the three masks mentioned above (line 15). Finally, we arrive at defining the hBlob object and derivation of centroid of TCP (lines 16–18). The results are shown in Figure 13.

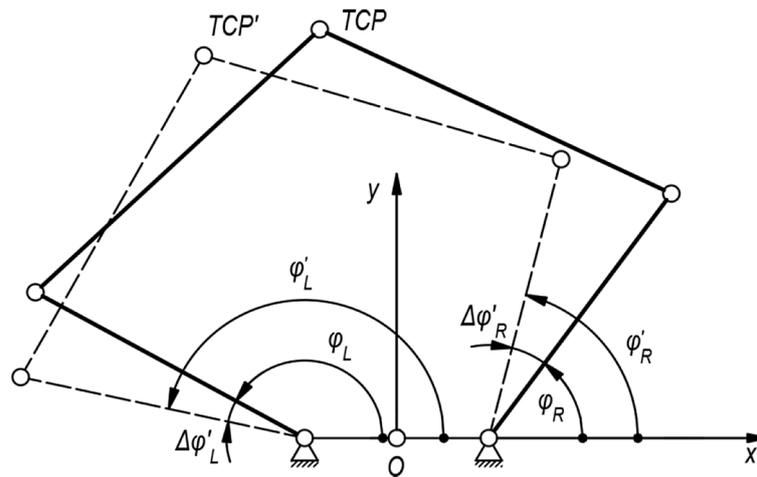


Figure 10. Vision-based correction of TCP position

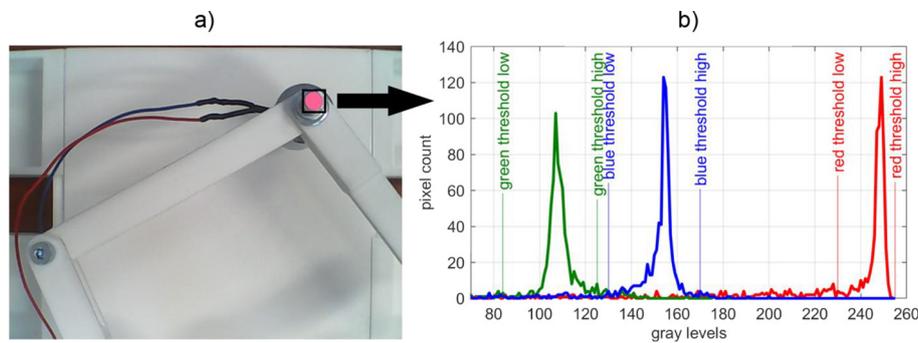


Figure 11. Coloured marking at tool centre point: a) captured image, b) histogram of marking

In order to get coordinates of TCP in millimeters one should perform additional calibration in the plane of TCP marking, thus additional  $px2xy\_det$  function would be needed. The mask-*Thresh* function used here is given by algorithm from Figure A.9 (Appendix). The principle is similar as in the case of defining the binary mask (Figure A.5 – Appendix), but here the high and low threshold values are used in conditional statement (line 5).

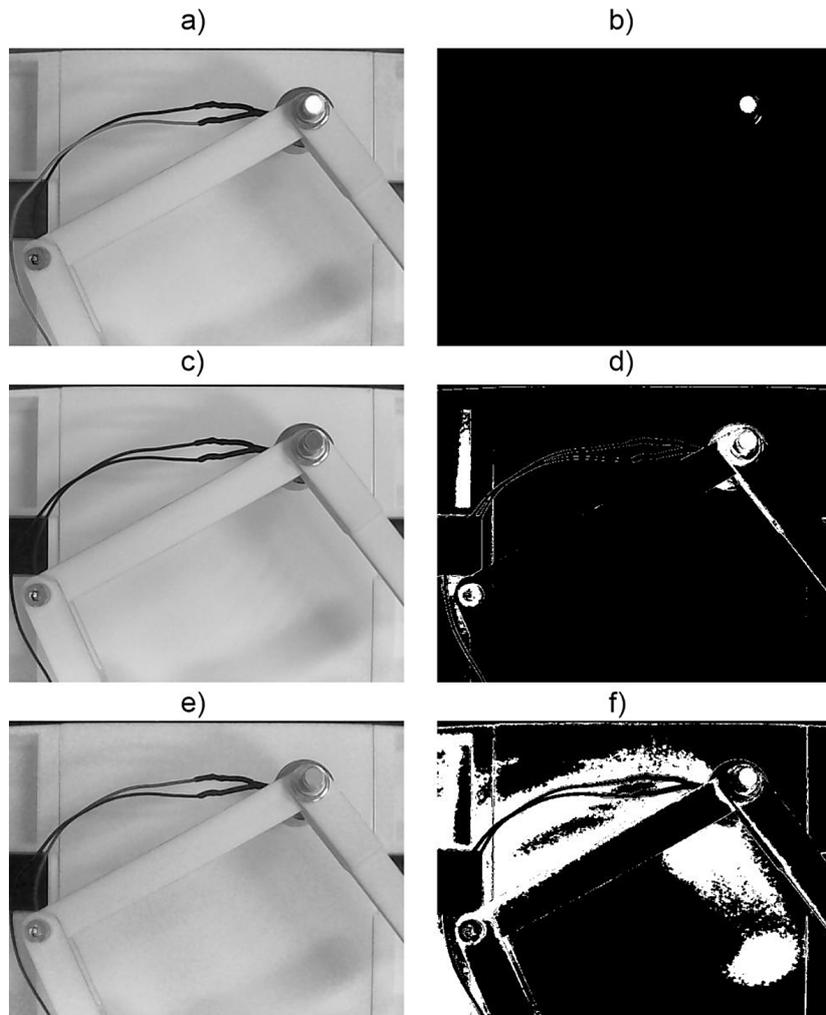
### Supervised classification of parts

Machine vision is commonly used in industrial settings to identify defective parts during the manufacturing process. These defective parts should be sorted using automated systems, such as specialized machines or robots. One effective method for fault detection is the use of neural networks trained under supervision. This section discusses data preprocessing, the architecture of convolutional neural network, and the results obtained for classification.

### Data preparation

As an example, we will consider small steel parts for automotive brakes after the galvanisation process. For some reason, the galvanisation not always is proper and a whole batch of products may be withdrawn by the client if only one has surface defects. Ideal for this kind of task are convolutional neural networks. Before we start building the classification model first we need to collect data for training, testing and validation. Images for the dataset were captured directly by the vision system of the manipulator. To automate data acquisition one may use an object detection algorithm (Figure A.5 from an Appendix) and trimming. As a result, we have collected 307 images of parts without defects (Figure 14a) and 307 defected (Figure 14b).

Next, to increase the number of images, data augmentation was performed. Each of the images was rotated about an angle of  $90^\circ$ ,  $180^\circ$  and  $270^\circ$ . This resulted in a total number of colour images of 2456 of size  $90 \times 90$  px



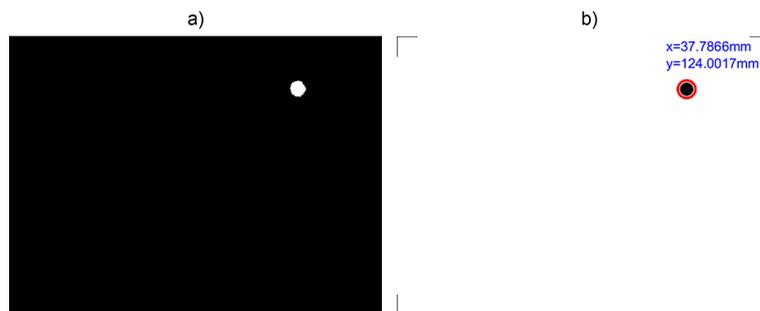
**Figure 12.** Image processing for colour detection: a) red band image, b) red band mask, c) green band image, d) green band mask, e) blue band image, f) blue band mask

each. The dataset was split into data for training (60%), validation (20%) and testing (20%) which gives respectively 1476, 490 and 490 images. In each of these subsets, the distribution of features (defect/no defect) is equal (50%/50%). Before training, images were subjected to pre-processing, namely zero-centering with respect

to the training dataset and converting into BGR images (Figure 15).

#### Deep convolutional neural network

For part classification task we came up with a deep convolutional neural network (CNN) with six hidden layers (Figure 16).



**Figure 13.** Results of the TCP detection algorithm: a) purple mask ( $M_p$ ), b) detected TCP



Figure 14. Image dataset: a) parts without defects, b) parts with defects

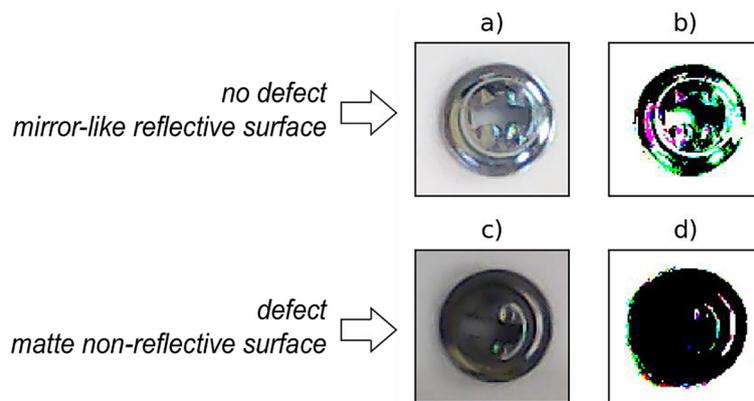


Figure 15. Image pre-processing: a) original image of part without defect, b) zero-centered BGR image of part without defect, c) original image of part with defect, d) zero-centered BGR image of part with defect

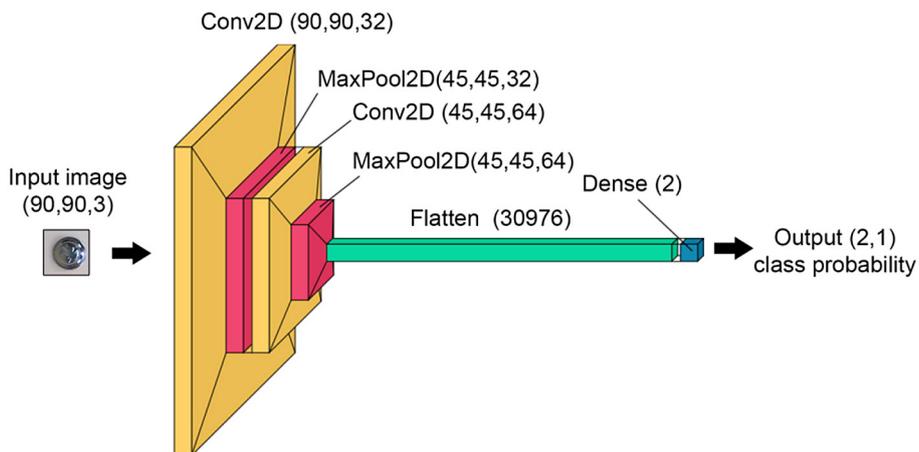


Figure 16. Convolutional neural network architecture for part classification

The network consists of two convolutional 2D layers with  $3 \times 3$  kernel and rectified linear unit activation function followed by max pooling layers with pool size  $2 \times 2$ . In the end, the output after pooling is flattened and returned after the dense layer with softmax activation as the probability

of class occurrence. The training was performed with Adam optimizer and as the loss the categorical cross-entropy was used. The results of testing are presented in Figure 17 as confusion matrix.

The CNN performed well, achieving high precision (0.98), recall (0.97) and F1-measure (0.97).

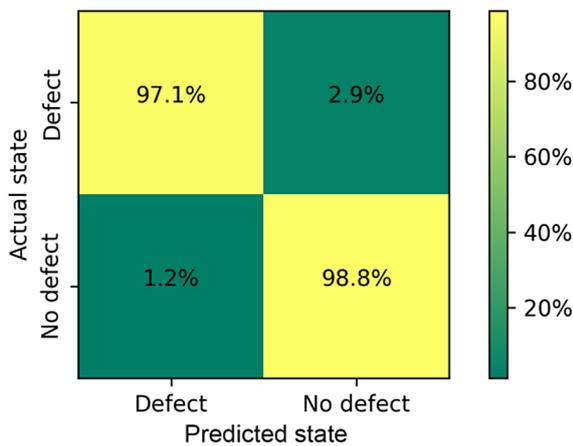


Figure 17. Confusion matrix for supervised classification

### Unsupervised classification of parts

Sometimes we encounter heavily imbalanced datasets, particularly when there is a limited amount of data representing the failure state. This situation is common in various diagnostic tasks where we lack extensive test results and primarily know how the system should perform under normal operating conditions. In such cases, it becomes necessary to employ unsupervised or semi-supervised learning approaches. This section discusses data preprocessing, autoencoder architecture, as well as training procedure and the resulting outcomes.

#### Data preparation

To simulate this situation we will use part of our dataset (Figure 14). For training and validation, we will use only images of parts without

defects. Testing will be performed with both defect and no defect images. This results in 738 (100% without defect) images for training, 245 (100% without defect) for validation and 490 (50% defective / 50% without defect) for testing. To decrease the training time we applied different pre-processing. Images were resized to  $28 \times 28$  px, transformed into greyscale and normalized.

#### Convolutional autoencoder

One of the typical applications of the unsupervised learning approach is anomaly detection. There are various anomaly detection methods, and one of them is to use of an autoencoder. An autoencoder is a special type of neural network designed to reconstruct its input. It might be used for dimensionality reduction and feature extraction. Input data are compressed by the encoder into the latent layer and then decoded by the decoder. Anomaly might be detected by analysis of latent space or reconstruction error. Figure 18 presents a convolutional autoencoder used for the detection of abnormal images in this study.

The encoder consists of five convolutional 2D layers with a rectified linear unit activation function and two max pooling layers. The decoder is constructed from five up-sampling transposed convolutional layers. While training, Adam optimizer with the loss as the mean squared error was used (Figure 19).

Training and validation loss converges to small value of approx. 0.15. The result of reconstruction of training data is presented in Figure 20.

The reconstructed images correspond to input ones but are blurred, which is typical for autoencoders.

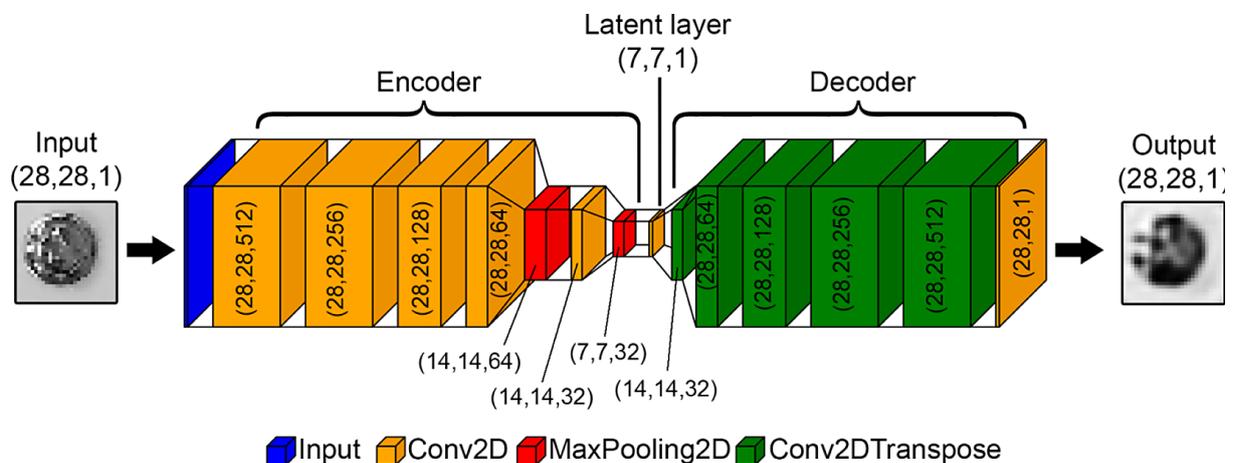


Figure 18. Convolutional autoencoder architecture

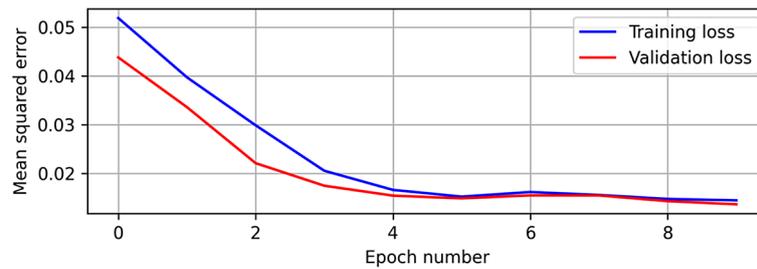


Figure 19. Autoencoder training and validation loss

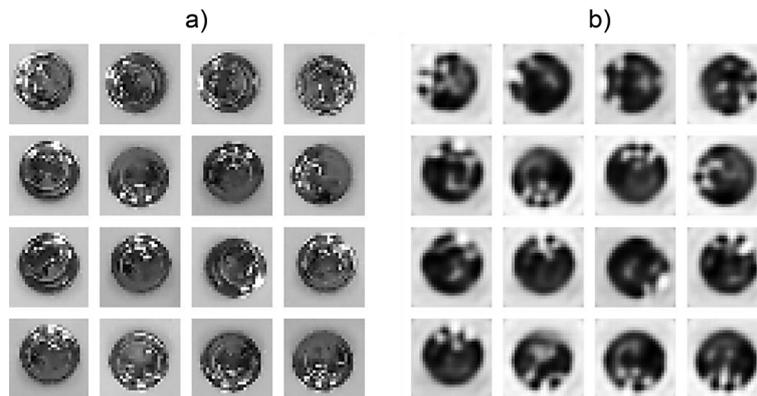


Figure 20. Training data reconstruction: a) input images, b) output images

### Latent space exploration

Figure 21 presents images compressed into the latent layer. It can be observed that the auto-encoder tries to learn the features of the image. In the case presented this feature corresponds to the gloss of the part which is expressed as brighter pixels. Parts with defects have a smaller number of those pixels.

To visualise the results we may transform latent space into three-dimensional space with the aid of T-distributed stochastic neighbour embedding (t-SNE) [34], which is shown in Figure 22.

After embedding the latent image data, a centred cluster corresponding to parts without defects is observed. Points for parts with defects are spread around this centre cluster. To detect whenever a point is within the blue cluster the k-means algorithm can be used. We assume that only blue points are known, which forms one cluster ( $k = 1$ ). Then we calculate the average of these points which corresponds to the cluster center. In the case presented center coordinates are near zero ( $t_{SNE1} = -0.15$ ,  $t_{SNE2} = 0.23$ ,  $t_{SNE3} = -0.04$ ). Next, we calculate the distance of each blue point to the cluster centre and average them, which gives us the limiting distance ( $r_{lim} = 17.58$ ). The point is considered an outlier if its distance to the cluster

centre is greater than  $r_{lim}$ . Otherwise, we assume that it is located within this cluster. The results of k-means-based classification are given in Figure 23 and in Figure 24 as the confusion matrix.

We achieve comparable metrics as in the case of the supervised learning approach. Exploration of a latent layer of an autoencoder gives high precision (1.00), recall (0.96) and f1-measure (0.98).

### Conclusions

This paper introduces a small, educational parallel SCARA robot designed for academic applications. The robot’s design and control framework are detailed, accompanied by example laboratory exercises to facilitate hands-on learning. Its low-cost and easy-to-assemble design enables equipping laboratories with multiple test stands, ensuring that each student or pair of students can actively engage with practical exercises. This approach promotes experiential learning in key areas such as robotics, machine vision, and neural networks.

The proposed teaching program is versatile and can be integrated into university courses in disciplines such as mechatronics and robotics, which demand interdisciplinary knowledge. Furthermore, the program can be expanded to

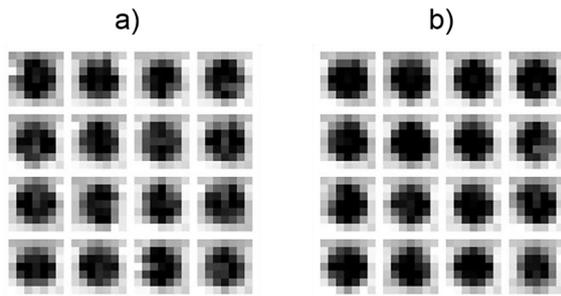


Figure 21. Test images after encoding: a) no defect, b) defect

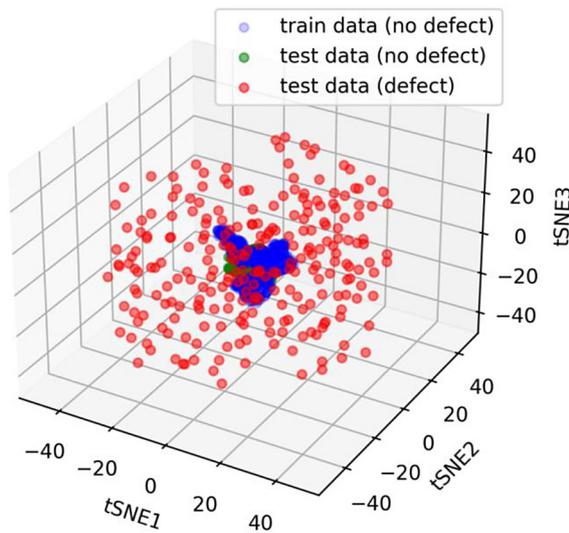


Figure 22. Latent space after dimensionality reduction by t-SNE

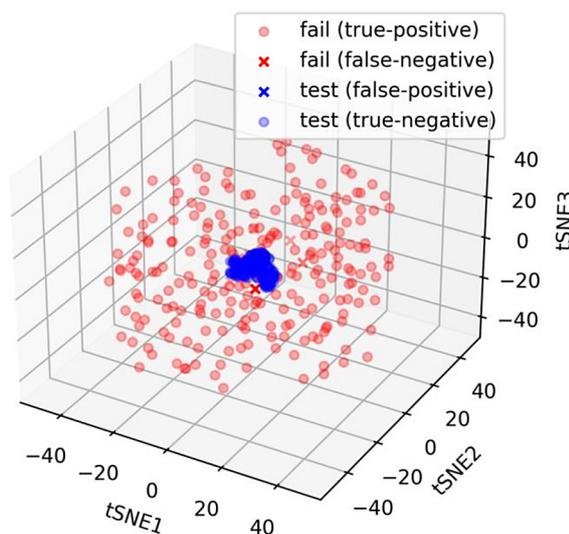


Figure 23. Outlier detection by k-means in embedded latent space of an autoencoder

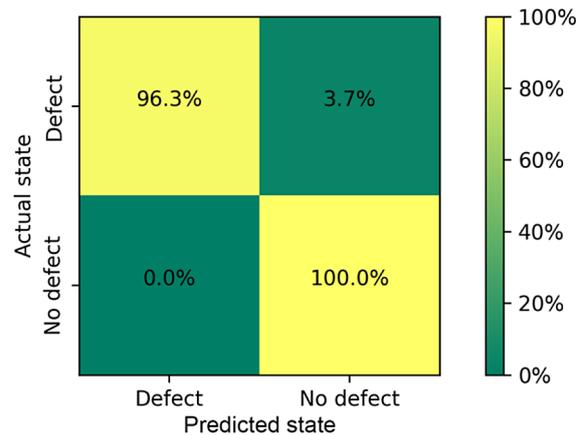


Figure 24. Confusion matrix for k-means classification of embedded latent space of an autoencoder

include rapid prototyping, combining robotic design with manufacturing techniques to enhance the scope of the project.

Significantly, the test stand replicates real-world industrial challenges, such as part classification and fault or anomaly detection, offering students practical insights into real-life applications. Notably, we investigate both supervised (using convolutional neural networks) and unsupervised (through autoencoder latent space exploration) approaches for classifying faulty parts. The unsupervised method achieved high performance, with a precision of 1.00, recall of 0.96, and an F1-measure of 0.98, which is comparable to the supervised approach that yielded a precision of 0.98, recall of 0.97, and an F1-measure of 0.97. This integration of theory and practice prepares students for addressing complex problems in industrial and research settings.

Future research could focus on alternative calibration methods, such as ArUco markers or QR codes, for pose estimation and 3D image reconstruction to detect the orientation, size and texture of spatial parts.

## REFERENCES

1. Grzejszczak T, Nocon W. Cooperative and individual planning strategies for autonomous robots in warehouse parcel transportation. *Advances in Science and Technology-Research Journal* 2025; 19: 406–18. <https://doi.org/10.12913/22998624/195453>
2. Boschetti G, Minto R. A Comparison of Control Strategies for Collaborative Mobile Robots. In: Rosati G, Gasparetto A, Ceccarelli M, editors.

- University of Padua, 2024; 165: 261–9. [https://doi.org/10.1007/978-3-031-67295-8\\_29](https://doi.org/10.1007/978-3-031-67295-8_29).
3. Gusan V, Titu M, Oprean C. Industrial robots versus collaborative robots - the place and role in nonconventional technologies. *Acta Technica Napocensis Series-Applied Mathematics Mechanics and Engineering* 2022; 65: 101–10.
  4. Mitsuoka R, Kubota T, Sato S, Ogawa K. Semi-autonomous touch method merging robot's autonomous touch and user-operated touch for improving user experience in robot touch. *Scientific Reports* 2024; 14. <https://doi.org/10.1038/s41598-024-67964-0>
  5. Burghardt A, Gierlak P, Skwarek W. Modeling of dynamics of cooperating wheeled mobile robots. *Journal of Theoretical and Applied Mechanics* 2021; 59: 649–59. <https://doi.org/10.15632/jtam-pl/141668>
  6. Buratowski T, Giergiel J, Uhl T, Burghardt A. The Kinematics Aspect of Robots Formation in Cooperation Tasks. In: Sasiadek J, editor. AGH University of Krakow, 2013; 121–8. [https://doi.org/10.1007/978-3-642-34020-8\\_10](https://doi.org/10.1007/978-3-642-34020-8_10)
  7. Szuster M, Hendzel Z, Burghardt A. Fuzzy Sensor-Based Navigation with Neural Tracking Control of the Wheeled Mobile Robot. In: Rutkowski L, Korytkowski M, Scherer R, Tadeusiewicz R, Zadeh L, Zurada J, editors. Rzeszow University of Technology, 2014; 8468: 302–13.
  8. Lee J, Park G, Ahn S, IEEE. A 2021; 1643–8. <https://doi.org/10.23919/ICCAS52745.2021.9649859>
  9. Andersson S, Granlund A, Hedelind M, Bruch J. Exploring the Capabilities of Industrial Collaborative Robot Applications. In: Safsten K, Elgh F, editors. Malardalen University, 2020; 13: 109–18. <https://doi.org/10.3233/ATDE200148>
  10. Ferraguti F, Pertosa A, Secchi C, Fantuzzi C, Bonfè M, IEEE. A Methodology for Comparative Analysis of Collaborative Robots for Industry 4.0. *Universita di Modena e Reggio Emilia*, 2019; 1070–5. <https://doi.org/10.23919/date.2019.8714830>
  11. Yamamoto T, Yaguchi H, Kato S, Okada H. Evaluation of impression difference of a domestic mobile manipulator with autonomous and/or remote control in fetch-and-carry tasks. *Advanced Robotics* 2020; 34: 1291–308. <https://doi.org/10.1080/01691864.2020.1780152>
  12. Rykala L, Typiak A, Typiak R, Kaunas Univ Technol. Tag Tracking with a Mecanum Wheeled Mobile Robot Using Ultra Wideband Signals. *Military University of Technology in Warsaw*, 2018; 1215–20.
  13. Xu J, Zhang M, Zhang J. Kinematic model identification of autonomous mobile robot using dynamical recurrent neural networks. In: Gu J, Liu P, editors. Harbin Engineering University, 2005; 1447–50.
  14. Burghardt A, Kurc K, Szybicki D, Muszynska M, Tutak J. The algorithm for determining the TCP point of a 2D scanner using a conical element. *Electronics* 2024; 13. <https://doi.org/10.3390/electronics13214292>
  15. Burghardt A, Szybicki D, Gierlak P, Kurc K, Pietrus P, Cygan R. Programming of industrial robots using virtual reality and digital twins. *Applied Sciences* 2020; 10. <https://doi.org/10.3390/app10020486>
  16. Burghardt A, Szybicki D, Kurc K, Muszynska M. Mechatronic designing and prototyping of a mobile wheeled robot driven by a microcontroller. *Journal of Theoretical and Applied Mechanics* 2020; 58: 127–42. <https://doi.org/10.15632/jtam-pl/115332>
  17. Burghardt A, Kurc K, Szybicki D, Muszynska M, Szczech T. Robot-operated inspection of aircraft engine turbine rotor guide vane segment geometry. *Tehnicki Vjesnik-Technical Gazette* 2017; 24: 345–8. <https://doi.org/10.17559/TV-20160820141242>
  18. Daneshjo N, Hajduová Z, Pajerská E, Danishjoo E. Specification of the application of vibrodiagnostics in assessing the state of the industrial robot. *Advances in Science and Technology-Research Journal* 2019; 13: 68–78. <https://doi.org/10.12913/22998624/103381>
  19. Helvoort J van, Steinbuch M, Lambrechts P, Molengraaf R van de. Analytical and experimental modelling for gain-scheduling of a double scara robot. *IFAC Proceedings Volumes* 2004; 37: 61–6. [https://doi.org/10.1016/S1474-6670\(17\)31081-9](https://doi.org/10.1016/S1474-6670(17)31081-9)
  20. Coman GC, Serban C, Ivan AM, Nicolescu AF. Virtual prototyping, physical structure development and PC control of a double arm (five link) SCARA robot. *IOP Conference Series: Materials Science and Engineering* 2019; 591: 012075. <https://doi.org/10.1088/1757-899X/591/1/012075>
  21. Siltala N, Vuola A, Heikkilä R, Tuokko R. A H-Scara Mini Robot - a Dual Parallel Kinematics Mini Manipulator. *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, 2010; 1–7.
  22. Vuola A, Heikkilä R, Prusi T, Remes M, Rokka P, Siltala N, Tuokko R. Miniaturization of Flexible Screwing Cell. In: Ratchev S, editor. *Precision Assembly Technologies and Systems*, Berlin, Heidelberg: Springer Berlin Heidelberg; 2010; 309–16.
  23. Salzmann J, Zheng G, Gerber N, Stieger C, Arnold A, Rohrer U, Nolte L-P, Caversaccio M, Weber S. Development of a miniature robot for hearing aid implantation. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009; 2149–54. <https://doi.org/10.1109/IROS.2009.5354305>
  24. Md. Hazrat A, Aizat K, Yerkan K, Zhandos T, Anuar O. Vision-based robot manipulator for industrial applications. *Procedia Computer Science* 2018; 133: 205–12. <https://doi.org/10.1016/j.procs.2018.07.025>

25. Mehta SS, Burks TF. Vision-based control of robotic manipulator for citrus harvesting. *Computers and Electronics in Agriculture* 2014; 102: 146–58. <https://doi.org/10.1016/j.compag.2014.01.003>
26. Stieber ME, McKay M, Vukovich G, Petriu E. Vision-based sensing and control for space robotics applications. *IEEE Transactions on Instrumentation and Measurement* 1999; 48: 807–12. <https://doi.org/10.1109/19.779178>
27. Wells G, Venaille C, Torras C. Vision-based robot positioning using neural networks. *Image and Vision Computing* 1996; 14: 715–32. [https://doi.org/10.1016/0262-8856\(96\)89022-6](https://doi.org/10.1016/0262-8856(96)89022-6)
28. Corke PI, Hager GD. Vision-based robot control. In: Siciliano B, Valavanis KP, editors. *Control Problems in Robotics and Automation*, Berlin, Heidelberg: Springer Berlin Heidelberg; 1998; 177–92.
29. Hashimoto K. A review on vision-based control of robot manipulators. *Advanced Robotics* 2003; 17: 969–91. <https://doi.org/10.1163/156855303322554382>
30. Bochen A, Ambrozkiwicz B. The influence of light intensity on the operation of vision system in collaborative robot. *Advances in Science and Technology-Research Journal* 2023; 17: 206–14. <https://doi.org/10.12913/22998624/169884>
31. Cader M, Oliwa R, Markowska O, Budzik G. Producing mobile robot gripper part prototypes from polymeric materials using additive manufacturing technology. Part I. Mechanical properties and material constants of specimens from acrylonitrile-butadiene-styrene copolymer. *POLIMERY* 2017; 62: 27–35. <https://doi.org/10.14314/polimery.2017.027>
32. Danielsson P-E, Seger O. Generalized and Separable Sobel Operators. In: Freeman H, editor. *Machine Vision for Three-Dimensional Scenes*, Academic Press; 1990; 347–79.
33. Zhang Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2000; 22: 1330–4. <https://doi.org/10.1109/34.888718>
34. Van der Maaten L, Hinton G. Visualizing data using t-SNE. *Journal of Machine Learning Research* 2008; 9: