ACTA TECHNICA JAURINENSIS
Vol. XX, No. Y, pp. ZZ-ZZ, 2026
10.14513/actatechjaur.00981

Research Article

Position and pose detection of electrical housings in industrial scenario with deep neural networks

Gábor Böcz[1], Dominik Wenesz[1], Dániel Molnár[1,*]

[1]ACSG Kft, Práter utca 5a, 9024 Győr, Hungary
*e-mail: daniel@acsg.hu

Submitted: 10/02/2026 Revised: 09/04/2026 Accepted: 20/04/2026 Published online: 12/06/2026

Abstract: Robotization, data-based decision-making, and machine vision-based process control or monitoring are now indispensable in industrial environments. In this article, we present our manufacturing cell development at ACSG Kft., which aimed to cost-effectively apply modern machine vision and data-based decision-making in the conversion of a machine suitable for large-volume series production to small-batch production. The result of our development is a modular system suitable for data-based decision-making using machine vision that requires minimal human intervention.

Keywords: machine vision; deep learning; industrial automation; synthetic data

I. INTRODUCTION

In recent years, deep neural networks, a branch of machine learning, have begun to develop and spread explosively. It is inevitable that such new technologies will eventually be widely used in industry, and in this context, this article aims to present the results and lessons learned from a development project carried out at a Hungarian SME over the past year.

The focus is on the application of neural networks (NN) in machine vision tasks, which can also provide solutions to problems in areas other than the case presented. During the research, the goal at ACSG Kft. in Győr was to reduce the changeover time and cost of an electronic connector housing machine with a fixed feeding path, which was successfully achieved at the end of the development.

The article details the modifications made to the machine, focusing on the application of neural networks and the results obtained.

II. PROBLEM DESCRIPTION

At ACSG Kft., electronic connectors were assembled manually, but an automatic machine was purchased earlier to increase productivity. The machine is optimized for high-volume production of a single type of product, but the company's typical production needs are for smaller volumes of products with high variance, in batches. The housing machine feeds the connectors to the housing step via a vibrating drum feeder, which must be replaced each time the product is changed, as each connector has unique dimensions. The cost of a changeover is estimated at between €5,000 and €8,000, not to mention the fact that the machine cannot be used during this time, which causes a further financial burden as it is taken out of production.

Due to the above-mentioned machine constraints, the development focused on automating the dosing system and the manufacturing process as much as possible. The goal was to upgrade the machine with modern, innovative hardware and software, while ensuring the stability required in the industry. An additional constraint during the project was that the expanded machine could not contain sensors that provide direct three-dimensional geometric data, i.e., point cloud-based data sets could not be received as software input.

With these goals and constraints in mind, the draft amendment was developed and the solution was designed and tested in a real-world environment.

III. SIMILAR INDUSTRIAL EXAMPLES

In today's industrial environment, machine vision systems for object detection, segmentation, and classification are commonplace, often connected to automation and robotics systems. Machine vision systems assist in quality assurance, inspection, production status monitoring, localization, and position detection tasks, which are performed using one or more two- or three-dimensional sensor data. Thus, they greatly assist in real-time process monitoring [1]. Both traditional and deep learning-based models are used in industry, either independently or in hybrid solutions. Systems using deep learning are often more adaptive than their traditional counterparts [2].

The most common applications in manufacturing processes are the detection of defective products, the identification of defective parts, the determination of the position and spatial rotation of objects, the classification of different types of raw materials, the automatic removal of raw materials from warehouses, and the detection of faulty assembly [1, 3]. In addition to the accuracy of the system output, these tasks also require fast interference, i.e., the system must be able to provide correct results in real time. In addition to traditional image processing techniques such as edge detection, corner detection, and geometric structure matching, convolutional neural networks are used as deep learning-based solutions [4].

Robotic automation systems supported by machine vision systems use two-dimensional or three-dimensional sensor data as input data, which are used in single or hybrid solutions in industrial environments [5]. Typically, systems based on combined sensor data provide a more robust solution, as the input data they provide covers a much larger part of the physical space than data from individually used sensors [5].

In the electrical housing industry, connectors are still mostly assembled by hand for small batches, while for high-volume production there is no need for an adaptive, easily changeable system. Leaving aside industry-specific characteristics and considering only the characteristics of the connectors, the task at hand is the spatial manipulation of large quantities of small, bulk objects with similar geometries. In industrial practice, robot manipulators with sensor-based control are often used for this purpose.

The vast majority of these systems use sensors capable of 2D or 3D imaging. In the case of high-precision positioning of bulk products, the required movement trajectory is typically calculated on the basis of a spatial point cloud or by fusing a point cloud with a two-dimensional image.

In addition to traditional image processing algorithms, deep learning-based approaches using convolutional or transformer neural networks have gained increasing prominence in research and industrial practice. These are used for quality assurance tasks [9, 10, 11], manufacturing process automation [12, 13], and simulation facilitation [14, 15, 16]. In addition, other machine learning-based decision models have also gained ground in industrial environments [17, 18, 19], reinforcing direct data-based decision-making.

Based on the findings of research into optimising licence plate recognition [20], a problem involving similar levels of noise, we drew the following conclusions regarding our own task. The input image may be noisy or blurry, therefore, we should use standard computer vision algorithms such as Wiener deconvolution, point spread function estimation, median filtering, bilateral filtering, morphological operations, and edge detection. In licence plate recognition, the region of interest is identified first, we adopted this approach for performance optimisation in our research.

Another similar application field is the detection of marine objects [21], which is also subject to noise and environmental conditions that can vary on a large scale. Fish, corals and underwater structures can be detected in real time despite the limited computing power available. In this study, the YOLOv4 head was modified to address the specific requirements of the problem. The researchers incorporated transformer self-attention-like features into the modification, which improved small object detection. Based on these findings, we concluded that modifying the head part of the YOLO models we use could optimise detections.

Despite the drawbacks of the 2D image-based solution, we opted for it because our aim was to develop a system that could be used for multiple industrial applications without requiring high-end hardware. This, of course, enforces constraints on the working area. In this case, the 3D position of the belt and the camera must be known relative to the robot coordinate system, as must the height of the connectors, which can be obtained from the CAD model. These can be calibrated and calculated relatively easily, even in noisy industrial environments. The detection system is significantly cheaper than most 3D vision-based systems.

IV. DIFFICULTIES

The following metrics were established for the product of the development to ensure its applicability in a real environment:

- At least 90% accuracy in the estimated and actual position of objects.
- Maximum of 5 working days for integration into a new connector system.
- Maximum 4 hours conversion time for connector types already included in the system.
- 0% mechanical damage due to trajectories generated by the system as output.
- A maximum time window of 500 ms for images received from one or more sensors, during which the system is able to generate trajectories and send them to the actuators.

During this research, the goal was to determine the trajectory from only 2D RGB images in a stable and industrial environment, which increases the complexity of the basic problem in several ways.

- The distance between the point located in real physical space and the sensor, as well as the angle of incidence relative to the sensor plane, cannot be directly determined from the image data, they are only calculated values.
- Industrial environments are often noisy and have changing lighting and visibility conditions.
- RGB cameras are highly dependent on lighting conditions.
- For bulk products, some features are partially or completely invisible.

Another goal for the system was to minimize hardware costs without drastically reducing efficiency. Therefore, the following tools were used during development:

- NVIDIA RTX 3060 Ti graphics card for training neural networks and generating synthetic data.
- Jetson Xavier NX 8GB for runtime environment.
- Raspberry Pi HQ Camera as the image sensor

These tools are commercially available, ensuring the project's wide adaptability.

V. INVESTIGATED METHODS

1. Standard computer vision and logic based solution

A classical computer vision approach combined with explicit rule-based logic was investigated for the detection and orientation estimation of small electrical housing components. In this method, each connector type was modeled individually, requiring the manual definition of geometric and logical rules specific to that component.

The pipeline typically consisted of image preprocessing steps such as grayscale conversion, noise reduction, and edge enhancement, followed by feature extraction based on low-level image primitives, including edges, corners, contours, and key geometric dimensions. Shape descriptors, distances between characteristic points, and angular relationships were used to identify connector types.

For orientation estimation, predefined reference features (e.g., corners, holes, or asymmetric edges) were detected, and their spatial relationships were evaluated using deterministic logic. Decision rules were implemented to infer the pose of the component based on these features. This approach required separate logic design for each connector variant, as changes in geometry or appearance necessitated new rules and parameter tuning.

2. Machine learning based solution

In addition to purely rule-based methods, a machine learning–assisted computer vision approach was explored. This method retained a similar overall structure to the classical approach but introduced supervised learning models to simplify or replace parts of the handcrafted logic.

Feature extraction was still required and typically relied on geometric, intensity-based, or shape-related descriptors derived from the image. These features were then used as input to traditional machine learning algorithms such as Random Forests, Support Vector Machines, or similar ensemble and classification models.

The trained models were used to classify connector types or assist in determining orientation based on learned decision boundaries rather than fully manual thresholds. While this reduced the complexity of some explicit rules, the approach still required connector-specific datasets, feature engineering, and model training. Each new connector geometry generally demanded retraining or adjustment of the feature set, making the method partially data-driven but still strongly dependent on manual design choices.

3. Deep learning based solution

A fully data-driven deep learning approach was also investigated, focusing primarily on convolutional neural network–based object detection models, with particular emphasis on the YOLO [6-8] (You Only Look Once) family of architectures.

In this method, detection and orientation-related information were learned end-to-end directly from annotated image data, without the need for explicit geometric rules or handcrafted feature extraction. The model internally learned hierarchical feature representations, ranging from low-level visual patterns to higher-level abstract concepts relevant to connector shape and orientation.

YOLO-based models were trained to localize components using bounding boxes and to associate orientation-related information implicitly through learned spatial patterns. The decision logic traditionally implemented in rule-based systems was instead encoded implicitly within the network’s learned parameters, allowing the system to generalize across variations in lighting, positioning, and minor geometric differences.

This approach relies primarily on dataset quality, annotation accuracy, and training configuration, rather than explicit algorithmic logic, making it fundamentally different from the previous two methods in terms of system design and development workflow.

VI. SYNTHETIC DATA GENERATION

Training neural networks properly requires a large amount of data, which is especially true in critical industrial environments. Training a model requires hundreds, and in some cases thousands, of labeled images to achieve the desired level of accuracy. It is also important to note that the images must cover most of the cases that occur in order to obtain a sufficiently generalized, not overtrained network that can function in a variable, noise-laden industrial environment [20].

In the case of connector housings, real images can be created in multiple positions, lighting conditions, and with varying backgrounds for each product, and then produced and labeled in a variety of ways. Individual parts of the process can be automated separately or together, but this requires significant hardware preparation. If labeling is also automated, i.e., the exact position of each connector in the image is known in the prepared image as well as in reality, then a level of rigidity appears in the system that is no longer capable of reproducing real production cells. In other words, in the case of full automation, some kind of actuator moves the connector, camera, and background, and the lighting is also controlled, which in practice results in an overly controlled environment in which the connectors cannot appear in bulk. During the research, we created and labeled 100 images for each of three different connector sizes in manual, partially automated, and fully automated modes. **Table 1** shows the average time required for the three connectors. In the first case, the operations were performed completely manually. In the second case, the position of the connectors was changed using a vibrating actuator. In the third case, the background and lighting were also automated. In the fourth case, the entire image creation process took place without human intervention, only the labeling was done manually. In the fifth case, labeling was also generated automatically, but this time the products were not displayed in bulk, instead, a six-degree-of-freedom robotic arm moved a connector in front of the camera.

Table 1. *Time consumption of data collection per image*

Type	*Capturing [sec]*	*Labeling [sec]*
Manual	38	310
Semi a. I	23	323
Semi a. II	17	312
Auto capture	6	334
Full auto	4	~0

The measurement data clearly shows that collecting this type of training data is time-consuming and labor-intensive. Fully automated data collection cannot generate labeled data for bulk products without human intervention.

Due to the aforementioned difficulties with real image training data, we explored other options, which led us to synthetically generated data. With this method, we use 3D models of the connectors to create photorealistic images with simulation and rendering software. Unlike their real-world counterparts, synthetic data does not require human input beyond the initial settings, the process is completely automatic. An additional advantage of this technique is that labeling is also performed on bulk products, as the software knows the position and orientation of all connectors relative to the camera. Another significant advantage is that environmental variables such as lighting, background, and camera lens specifications can be freely adjusted. With this method, the set of training data can be expanded beyond real-world conditions, making it easy to cover rare compositions, such as when an unwanted object enters the workspace or the connectors are in an unstable position relative to each other. For the same connectors, **Table 2** shows the times required to generate an average of 100 images, depending on the resolution of the images.

Table 2. *Synthetic data generation per image*

Resolution	*Capturing [sec]*	*Labeling [sec]*
640x480	11	~0
1280x720	26	~0
1920x1080	42	~0
2560x1440	64	~0

The synthetically generated data requires much less time than data created in a real environment. However, it is important to note that if neural networks are trained using artificially generated data, they will almost always achieve poorer accuracy on real images than if they had been trained on the same number of real images. This difference can be reduced with the appropriate simulation and rendering settings, which we will discuss in more detail in the results section.

Training neural networks or creating algorithms on synthetic data is usually challenging due to the gap between the synthetic and real domains. This means that models trained on synthetic data alone tend to perform poorly in real-world scenarios. To reduce this gap, we used photorealistic rendering and surface textures. These were randomly selected for each connector, the other objects, and the background. By extending the synthetic domain in this way, we produced training data that was successfully used to train models capable of operating in real-world scenarios with the aforementioned workspace constraints.

VII. Results by vision method

To compare the results of each method, we conducted real-world testing of the pick-and-place part of the system, as we aimed to test each method and algorithm in a real industrial environment. Performance was measured based on the following quantities: the inference speed of detection, the number of successful and failed pick-ups and placements by the robot, the time taken to configure for a new connector type, and the time taken by humans to configure a new connector. The success rate was calculated based on all the pick and place cycles. These parameters were measured throughout our research. For each method, we display the best results and discuss the overall conclusion.

1. Standard computer vision and logic based solution

Traditional computer vision and rule-based methods performed poorly when used to detect small electrical housing components and estimate their orientation, and their reliability was limited across test cases. A key limitation of this method was the heavy manual workload, since each connector type required a separate set of rules, parameters, and geometric definitions. As a result, setting up the configuration took considerable time and required substantial engineering work each time a new connector variant was introduced.

The produced software was sensitive to environmental conditions, including changes in lighting, differences in the background, and small shifts in component placement. Small variations from the expected visual features often caused false detections or errors in the estimated pose. The method struggled to handle several connector types at once, especially when the parts looked similar, because the hand-crafted rules were not specific enough to separate them consistently.

Performance declined even more when the system encountered connectors in bulk, as occlusions, overlapping components, and various orientations were common in these cases. The logic based method could not keep feature extraction consistent or apply rules in a stable way, which led to unpredictable behavior, which was unacceptable in industrial applications. The results of the most effective logic-based model are presented in **Table 3.**

Table 3. *Best performing logic based solution*

Performance metric	*Value*
Inference speed per connector	90 ms
Successful pickup	44 %
Successful place	34 %
Human time per configuration	~ 8 h
New connector model creation time	~ 12 h

2. Machine learning based solution

The machine learning–assisted computer vision approach produced moderately improved performance compared to the purely rule-based solution. The use of supervised learning models enabled more flexible decision boundaries and reduced the reliance on rigid threshold-based logic, resulting in slightly higher detection stability under controlled conditions.

However, many of the limitations observed in the classical approach persisted. The system continued to require connector-specific feature engineering and dataset preparation, leading to considerable development effort for each new component type. Sensitivity to environmental variations, such as lighting changes and visual noise, remained a notable issue, although to a lesser extent than in the purely logic-based method.

Similar to the standard computer vision approach, the machine learning solution showed reduced reliability when handling visually similar connector types simultaneously and when processing bulk-arranged components. Occlusions and ambiguous feature representations negatively impacted classification and orientation estimation, indicating that the learned models were still constrained by the quality and discriminative power of the handcrafted input features. **Table 4** shows the data regarding the best machine learning based model.

Table 4. *Best performing machine learning based solution*

Performance metric	*Value*
Inference speed per connector	210 ms
Successful pickup	62 %
Successful place	47 %
Human time per configuration	~ 4 h
New connector model creation time	~ 7 h

3. Deep learning based solution

Using the methodology based on neural networks, we achieved much better results in our research than with the previous two methods. Several network architectures were evaluated, of which the YOLO models proved to be the most effective for the task. The YOLO architecture is a single-stage convolutional neural network-based architecture that performs feature extraction, bounding box regression, and classification within a single step on a single network. Furthermore, with a suitable detector head, it can also perform keypoint regression, which allows the exact orientation of each object to be determined. The model divides the input image into grids, along which probability scores are assigned to the objects detected within the grid, and then the final output is obtained from the interpolation of these, which contains the class, bounding box, probability score, and confidence score, as well as the data associated with the keypoints for a given network. Due to its architecture, it provides a completely data-driven solution.

YOLO models are available with varying numbers of parameters. Networks with larger parameters typically require more data for training, have a slower training process, and require more powerful hardware to run interference. In order to ensure that the developed system is cost-effective and can be easily implemented across the industry, we used networks with the lowest hardware requirements, i.e., the small and nano models. The results for the best deep learning based model can be seen in **Table 5.**

Table 5. *Best performing machine learning based solution*

Performance metric	***Value***
Inference speed per connector	110 ms
Successful pickup	97 %
Successful place	96 %
Human time per configuration	~ 0.2 h
New connector model creation time	~ 13 h

VIII. Final system

During the research, the final solution included mechanical modifications and a complete machine vision system with the hardware specifications established at the beginning of the research.

From a mechanical point of view, the vibrating drum feeder was removed from the original machine and replaced by a STAUBLI TS40 SCARA robot and a conveyor. The robot follows the trajectory generated by the machine vision system to pick up the bulk connectors moving on the conveyor belt in the work area and place them in the nest on the JAM, from where the machine performs the housing process without modification. The robot cell has been designed in accordance with the relevant safety standards.

The machine vision system can be trained completely offline using only CAD models. The only step in the training process that requires human intervention is specifying the correct insertion position and gripping points of the connector in the CAD model coordinate system. The system then generates virtual workspaces, randomly places the connectors in them, and generates training images and labels for tuning the neural networks (**Fig. 1**). Next, a YOLOv8n-based model used for object detection is trained to detect connectors on a conveyor belt, and another YOLOv8n-based model is trained to detect keypoints associated with the connectors. The synthetic data learned networks are then validated and tested by the system, and if their results meet the set compliance metrics, they are uploaded to the edge device in the robot cell and production can start for the newly added connector.

The robot and robot cell hardware cost around 50,000 euros. The computer vision system hardware, including the processing unit, cameras, lenses, and the computer used to generate synthetic data and train models, cost 3,000 euros. This brings the total hardware cost of the system to 53,000 euros, which would have covered four reconfigurations of the original system. It is important to note that with the original system, there was a recurring cost for each reconfiguration, even if the configuration had been used previously. Not to mention that the parts used for reconfiguration had to be stored in a warehouse, taking up at least one pallet. In contrast, the new system incurs only a utility cost for training the model, which is negligible compared to the original system's expenses. The new system's reconfiguration time is one working day, including training the models and the mechanical configuration. The training can run while the machine is still operating, reducing downtime to one hour per reconfiguration, this was seven days with the old system. At ACSG Kft., we use hundreds of connector types with multiple pin numbers, bringing the total connectors used to the thousands. Processing this many connectors was impossible with the old system.

The object detection network has not been modified from the original YOLOv8n model, but the keypoint detection network has been given several new detection heads, which, in addition to the keypoints, also provide the connectivity, orientation, visibility, and correctness (i.e., error-free) of each connector as a scalar head output in the output tensor. Based on the values of this tensor, the trajectory through which the robot can pick up the given connector is generated for each connector. In addition to the two networks, a third network monitors the safety of the workspace so that if it detects a foreign, inappropriate object, it stops the robot and sets the system to error mode.

Figure 1. *Synthetic training data*

The system takes continuous pictures of the conveyor belt and processes them. First, the entire image is fed through a YOLO model, which classifies and locates the objects on the conveyor belt (**Fig. 2**). Connectors belonging to the currently processed type are labelled as 'front', 'back', 'side' or 'unprocessable', all other connectors or objects are ignored (background class) or labelled as 'other entity' (changeable parameter). The system then locates the keypoints on the pickable connectors within the working area (**Fig. 3**), for this purpose, a smaller YOLO model is used. Based on the configuration parameters, the system selects a connector, which the SCARA robot then picks up and places in the nest for further processing. If the manipulation fails for any reason, the system checks if the robot arm is obstructing the camera view. If so, it moves the arm out of the working zone and processes a new image. If the pick-up process succeeds, the system decides whether to move the conveyor belt to feed new connectors into the working area, thus maintaining the cycle time. Once the conveyor belt has been moved and the arm is outside the camera's view, the detection process runs again. For safety reasons, a third YOLO model processes lower-quality images from the working area and the nest area to detect any malfunctions or potentially dangerous activities (e.g. a human hand in the working zone). If there are no processable connectors on the conveyor belt, the system moves the belt until there are, and also notifies the operator.

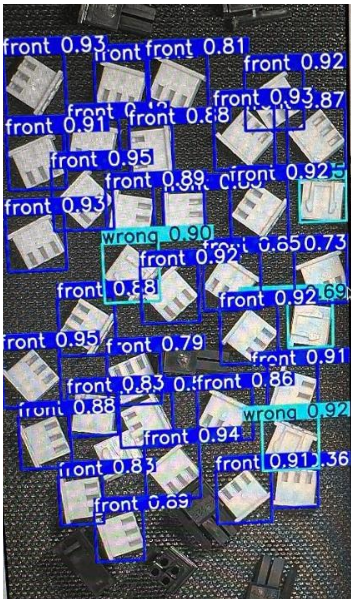

Figure 2. *Detection on conveyor*

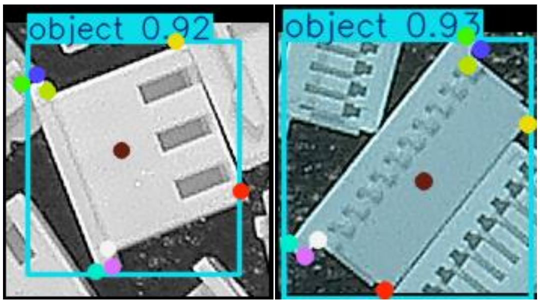

Figure 3. *Keypoint detection on two connector types*

The system developed during the project is fully modular and can also be used in other areas where fast transfer times are required in machine vision-controlled production cells in a cost-effective manner.

Acknowledgement

The project was made possible by the GINOP_PLUSZ-2.1.1-21-2022-00035 grant program, with the support of the European Union and the Hungarian government. We would like to take this opportunity to express our gratitude!

AUTHOR CONTRIBUTIONS

G. Böcz: Conceptualization, Theoretical analysis.

D. Molnár: Field testing, Supervision, Review and editing.

D. Wenesz: Model training and testing.

Disclosure statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

ORCID

G. Böcz http://orcid.org/0000-0000-0000-0000

References

[1] R. Kühlechner, "Object detection survey for industrial applications with focus on quality control," *Prod. Eng. Res. Devel.*, vol. 19, pp. 1271–1291, 2025. https://doi.org/10.1007/s11740-025-01369-4

[2] Y. Ma, J. Yin, F. Huang, et al., "Surface defect inspection of industrial products with object detection deep networks: a systematic review," *Artif. Intell. Rev.*, vol. 57, no. 333, 2024. https://doi.org/10.1007/s10462-024-10956-3

[3] D. Y. Patrashko and V. Gurau, "Machine Learning-Powered Vision for Robotic Inspection in Manufacturing: A Review," *Sensors*, vol. 26, no. 3, p. 788, 2026. https://doi.org/10.3390/s26030788

[4] Z. Qi, L. Ding, X. Li, J. Hu, B. Lyu, and A. Xiang, "Detecting and classifying defective products in images using YOLO," *arXiv preprint arXiv:2412.16935*, 2024. https://doi.org/10.48550/arXiv.2412.16935

[5] N. Mahjourian and V. Nguyen, "Multimodal object detection using depth and image data for manufacturing parts," *arXiv preprint arXiv:2412.09062*, 2024. https://doi.org/10.48550/arXiv.2412.09062
[6] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: a review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 11, pp. 3212-3232, 2019. https://doi.org/10.1109/TNNLS.2018.2876865
[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779-788. https://doi.org/10.1109/CVPR.2016.91
[8] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *preprint*, 2017. https://doi.org//10.13140/2.1.3104.1927
[9] J. Zender, S. Maier, A. Herkommer, and M. Layh, "Synthetic Data Generation for AI-Based Quality Inspection of Laser Welds in Lithium-Ion Batteries," *Sensors*, vol. 25, no. 23, p. 7301, 2025. https://doi.org/10.3390/s25237301
[10] G. López, P. D. Ramírez, E. Vega, F. Pizarro, J. Toro, and C. Parra, "WeldVGG: A VGG-Inspired Deep Learning Model for Weld Defect Classification from Radiographic Images with Visual Interpretability," *Sensors*, vol. 25, no. 19, p. 6183, 2025. https://doi.org/10.3390/s25196183
[11] S. Are, N. Krishna, J. Thati, and K. Gopi, "Federated multi scale vision transformer with adaptive client aggregation for industrial defect detection," *Sci. Rep.*, vol. 15, 2025. https://doi.org/10.31181/sdmap21202512
[12] M. Parikh, S. Ramavath, P. Shanmugasundaram, R. Handaragal, and R. Nathiya, "Deep Learning for Automated Defect Detection in Industrial Manufacturing," in *Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2025. https://doi.org/10.1109/ICESC65114.2025.11212369
[13] D. Miller, et al., "Enhanced 6D pose estimation for robotic assembly using hybrid deep learning," *Robotics and Computer-Integrated Manufacturing*, vol. 91, pp. 102-115, 2025. https://doi.org/10.1016/j.robot.2021.103775
[14] T. Koller and C. Tóth-Nagy, "Application of neural networks in vehicle simulation as a substitute for driver models," *Cognitive Sustainability*, vol. 4, no. 2, 2025. https://doi.org/10.55343/CogSust.142
[15] F. Nikolić and M. Čanađija, "Machine learning of structure – property relationships: an application to heat generation during plastic deformation," *Facta Universitatis, Series: Mechanical Engineering*, vol. 23, no. 4, pp. 687-707, 2025. https://doi.org/10.22190/FUME240215019N
[16] I. Jang and Y. Jang, "Prediction of contact distribution on rough surfaces using deep learning algorithms," *Facta Universitatis, Series: Mechanical Engineering*, vol. 23, no. 4, pp. 757-785, 2025. https://doi.org/10.22190/FUME250307026J
[17] M. Sandra, C. Nishanthini, S. Naryanamoorthy, and N. Almakayeel, "A smart decision framework for sustainable management of C&D waste using picture fuzzy decision model," *Spectrum of Mechanical Engineering and Operational Research*, vol. 2, no. 1, pp. 130–146, 2025. https://doi.org/10.31181/smeor21202533
[18] H. Fazlollahtabar, "Optimizing Robotic Manufacturing in Industry 4.0: A Hybrid Fuzzy Neural Bayesian Belief Networks," *Spectrum of Mechanical Engineering and Operational Research*, vol. 2, no. 1, pp. 191-203, 2025. https://doi.org/10.31181/smeor21202543
[19] A. Hussain and K. Ullah, "An Intelligent Decision Support System for Spherical Fuzzy Sugeno-Weber Aggregation Operators and Real-Life Applications," *Spectrum of Mechanical Engineering and Operational Research*, vol. 1, no. 1, pp. 177-188, 2024. https://doi.org/10.31181/smeor11202415
[20] J. Schmidt and T. Klein, "Leveraging diffusion models for synthetic data in low-volume manufacturing," *Applied Soft Computing*, vol. 142, pp. 110-125, 2025.
[21] K. S. Ravichandran, "Estimation of Automatic License Plate Recognition Using Deep Learning Algorithms," *Spectrum of Decision Making and Applications*, vol. 2, no. 1, pp. 100-119, 2025. https://doi.org/10.31181/sdmap21202512
[22] M. D. Baig and H. B. U. Haq, "Marine Object Detection using YOLOv4 Adapted Convolutional Neural Network," *Decision Making Advances*, vol. 2, no. 1, pp. 83-91, 2024. https://www.dma-journal.org/index.php/dema/article/view/28/19