

Article

Bin-Picking Solution for Randomly Placed Automotive Connectors Based on Machine Learning Techniques

Pedro Torres ^{1,2,*}, Janis Arents ³, Hugo Marques ¹ and Paulo Marques ^{1,4}

¹ Instituto Politécnico de Castelo Branco, 6000-084 Castelo Branco, Portugal; hugo@ipcb.pt (H.M.); paulo.marques@stshield.com (P.M.)

² SYSTEC—Research Center for Systems & Technologies, 4200-465 Porto, Portugal

³ Institute of Electronics and Computer Science, LV-1006 Riga, Latvia; janis.arents@edi.lv

⁴ StoneShield—Engineering, Lda, 6000-790 Castelo Branco, Portugal

* Correspondence: pedrotorres@ipcb.pt

Abstract: This paper presents the development of a bin-picking solution based on low-cost vision systems for the manipulation of automotive electrical connectors using machine learning techniques. The automotive sector has always been in a state of constant growth and change, which also implies constant challenges in the wire harnesses sector, and the emerging growth of electric cars is proof of this and represents a challenge for the industry. Traditionally, this sector is based on strong human work manufacturing and the need arises to make the digital transition, supported in the context of Industry 4.0, allowing the automation of processes and freeing operators for other activities with more added value. Depending on the car model and its feature packs, a connector can interface with a different number of wires, but the connector holes are the same. Holes not connected with wires need to be sealed, mainly to guarantee the tightness of the cable. Seals are inserted manually or, more recently, through robotic stations. Due to the huge variety of references and connector configurations, layout errors sometimes occur during seal insertion due to changed references or problems with the seal insertion machine. Consequently, faulty connectors are dumped into boxes, piling up different types of references. These connectors are not trash and need to be reused. This article proposes a bin-picking solution for classification, selection and separation, using a two-finger gripper, of these connectors for reuse in a new operation of removal and insertion of seals. Connectors are identified through a 3D vision system, consisting of an Intel RealSense camera for object depth information and the YOLOv5 algorithm for object classification. The advantage of this approach over other solutions is the ability to accurately detect and grasp small objects through a low-cost 3D camera even when the image resolution is low, benefiting from the power of machine learning algorithms.

Keywords: bin-picking; machine learning; robotics; YOLOv5; Industry 4.0

Citation: Torres, P.; Arents, J.; Marques, H.; Marques, P. Bin-Picking Solution for Randomly Placed Automotive Connectors Based on Machine Learning Techniques. *Electronics* **2022**, *11*, 476. <https://doi.org/10.3390/electronics11030476>

Academic Editor: Gemma Piella

Received: 11 January 2022

Accepted: 3 February 2022

Published: 6 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Robotic machine manufacturers targeting automatic assembly processes are being pushed to do more and better with less waste, urged to align towards the UN Sustainable Development Goal 12.5 (to substantially reduce waste generation through prevention, reduction, recycling and reuse, by 2030) [1]. Automatic assembly is a complex process that involves strict quality control, mostly by means of computer vision. If an assembled object is faulty, it is rejected and may end up in the trash. The work described in this paper describes an application that classifies, collects, sorts and aligns different objects for later reuse. At the input, the system receives boxes of faulty assembled connectors (different sizes, shapes and colors, and in different poses), while at the output, different boxes are filled with aligned and same-type connectors, as explained later in Section 3.

The automobile industry has always imposed the growth of the cable assembly industry and is characterized by many technological changes in a short period of time. Vehicles are becoming more comfortable, safer, more efficient and less polluting, but they are also increasingly complex systems with lots of electronics. The Electric Distribution System (EDS) has to constantly adapt to these changes in terms of concept quality and technological requirements. According to recent market reports [2,3], the rise of electric vehicles is driving the market. In 2020, global sales of plug-in electric cars increased 39% from the previous year to 3.1 million units. By the end of 2026, annual sales of battery-powered electric cars are expected to exceed 7 million and to contribute about 15% of total vehicle sales. This increase in sales is mainly due to increased regulatory standards imposed by various organizations and governments to limit emissions and promote zero-emission automobiles. As more electric vehicles circulate, the electric harness market is also expected to witness growth, since electric harnesses are used more in electric vehicles than in conventional fossil fuel vehicles.

To meet the growing needs, the electric harness market needs to digitize and automate processes to increase production levels and also reduce the number of failures, often associated with human error. The tasks performed in wire harnesses have traditionally been difficult for robots. Therefore, the solution involves changing the harness architectures, through a Design-for-Automation logic, as well as automating some current processes through robotic stations.

Traditionally, grasping and sorting randomly positioned objects requires human resources, which is a very monotonous task, lacks creativity and is no longer sustainable in the context of smart manufacturing [4]. Industrial robots, however, require a supplementary cognitive sensing system that can acquire and process information about the environment and guide the robot to grasp arbitrarily placed objects out of the bin. In industry settings, this problem has been commonly referred to as bin-picking [5] and also historically addressed as one of the greatest robotic challenges in manufacturing automation [6]. Bin-picking depends on visual-based robotic grasping approaches, which can be divided into methods where the shape of the object is analyzed (analytic approaches) or machine learning-based methods (data-driven approaches). Data-driven approaches can be categorized as model-free or model-based, where model-based approaches require prior knowledge of the object to determine the grasping position and model-free methods directly search for possible grasping points [7]. In the process of sorting automotive connectors, several different object types can be present and mixed in one pile. To be efficient, it is crucial to determine the object type before grasping as different grasping approaches are required for different connectors. Analytic methods fall short due to the high level of diversity in the region of interest. However, machine learning approaches tend to generalize and cope with uncertainties of the environment. Therefore, in this article, we focus on model-based, machine learning grasping methods.

The remainder of this paper is organized as follows: Section 2 describes the state-of-the-art and related works. Section 3 describes the materials and methods for the bin-picking solution. Section 4 presents the experimental results achieved and, finally, Section 5 presents the conclusions and future work.

2. Related Work

Bin-picking is a methodology used in Vision-Guided Robotics systems in which pieces are randomly selected and extracted in a container, using a vision system for location and a robotic system for extraction and its subsequent replacement. In recent years, a large number of 3D vision systems have emerged on the market that make it possible to implement bin-picking solutions in a smart factory context. Photoneo [8] is one of these brands which provides a 3D vision system, with software capable of training Convolutional Neural Networks (CNN) to recognize and classify objects and integrate with different models of robots. In addition, several other players bring machine vision solutions to the market for bin-picking applications, including Zivid [9], Solomon [10],

Pickit [11] and more. All of these systems provide very efficient and robust features for the industry, but they are still very expensive systems and are not accessible to the vast majority of small- and medium-sized enterprises (SME). This has led to the pursuit of alternative solutions based on more low-cost 3D vision cameras, investing in the research and the improvements of the machine learning algorithms. One such solution is proposed in [12], where the authors propose an object detection method based on the YOLOv5 algorithm, which can perform accurate positioning and recognition of objects to be grasped by an arm robot with an Intel RealSense D415 camera in an eye-to-hand configuration.

Bin-picking solutions have been studied for a long time, and in [13], some limitations and challenges of current solutions for the industry are identified and a system for grasping sheet metal parts is proposed. In [14], a solution is proposed with an ABB IRB2400 robot with a 3D vision system for picking and placing randomly located pieces. More recently, in [15] the authors propose a CAD-based 6-DoF (degree of freedom) pose estimation pipeline for robotic random bin-picking tasks using the 3D camera.

Picking only one object in a pile of random objects is a very challenging task, and in [16] a method is proposed to first compute grasping pose candidates by using the graspability index. Then, a CNN is trained to predict whether or not the robot can pick one and only one object from the bin. In [17], an approach for bin-picking industrial parts with arbitrary geometries is proposed based on the YOLOv3 algorithm. In [18], a flexible system for the integration of 3D computer vision and artificial intelligence solutions with industrial robots is proposed using the ROS framework, a Kinect V2 sensor and the UR5 collaborative robot.

One of the challenging tasks in bin-picking systems is identifying the best way to grip an object, therefore, it is necessary to identify the best gripper for the operation in addition to locating the objects and calculating the pose. In [19], the authors propose a system for the detection of object location, pose estimation, distance measurement and surface orientation angle detection. In [20], an object pose estimation method based on a landmark feature is proposed to estimate the rotation angle of the object. The sensitivity of the 3D vision system is very important to the success rate of a bin-picking solution; obviously, low-cost vision systems are useful for demonstrating concepts, but they are not usually suitable for working day-to-day in industrial scenarios.

The success rate of the bin-picking solution beyond the vision sensor depends a lot on the efficiency of the implemented algorithms. In [21], the authors compared the results of point cloud registration based on ICP (Iterative Closest Point) with data from different 3D sensors to analyze the success rate in bin-picking solutions. Object detection is one of the main tasks of computer vision, which consists of determining the location in the image where certain objects are present, as well as classifying them. The rapid advances of machine learning and deep learning techniques have greatly accelerated the achievements in object detection. With deep learning networks and the computing power of GPUs, the performance of object detectors and trackers has greatly improved. In [22], a review of object detection methods with deep learning is performed, where the fundamentals of this technique are discussed. One of the algorithms that have emerged in recent years is YOLO (You Only Look Once). The YOLO model is designed to encompass an architecture that processes all image resources (the authors called it the Darknet architecture) and is followed by two fully connected layers performing bounding box prediction for objects. Since its inception in 2015, YOLO has evolved, and in 2020 the company Ultralytics converted the previous version of YOLO into the PyTorch framework, giving it more visibility. YOLOv5 is written in Python instead of C as in previous versions. In addition, the PyTorch community is also larger than the Darknet community, which means that PyTorch will receive more contributions and growth potential in the future. The complete study of the evolution of YOLO can be found in [23]. YOLOv5 is now a reference and is extensively used in object detection tasks in various domains. As an example, in [24] it is used as a face detection algorithm suitable for complex scenes, and in [25] it is used as a

real-time detection algorithm for kiwifruit defects. In this work, YOLOv5 is the algorithm implemented for the identification and recognition of electrical connectors in a bin-picking application.

3. Materials and Methods

This section describes the methodology followed to implement our bin-picking solution for small automotive connectors and the machine learning algorithm used for object recognition and the respective robot navigation process. The core concept for our approach is depicted in Figure 1.

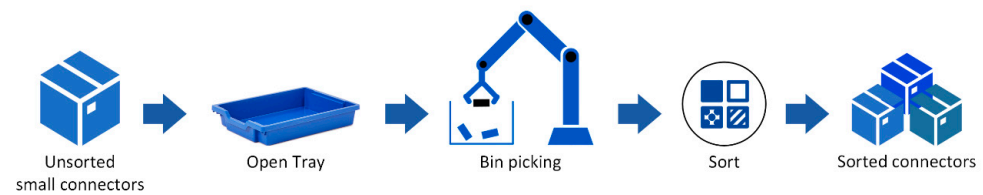


Figure 1. Bin-picking concept applied to unsorted small plastic connectors.

The process of assembling seals into connectors may produce a significant amount of poorly assembled connectors. The connectors that fail the quality tests are placed in large boxes for reuse. In the end, each box will contain multiple types of unsorted connectors, as depicted in Figure 2a. Each box is then verted (still unsorted) in open trays and our goal is for the robot to perform the bin-picking process and sort the connectors into different output boxes.

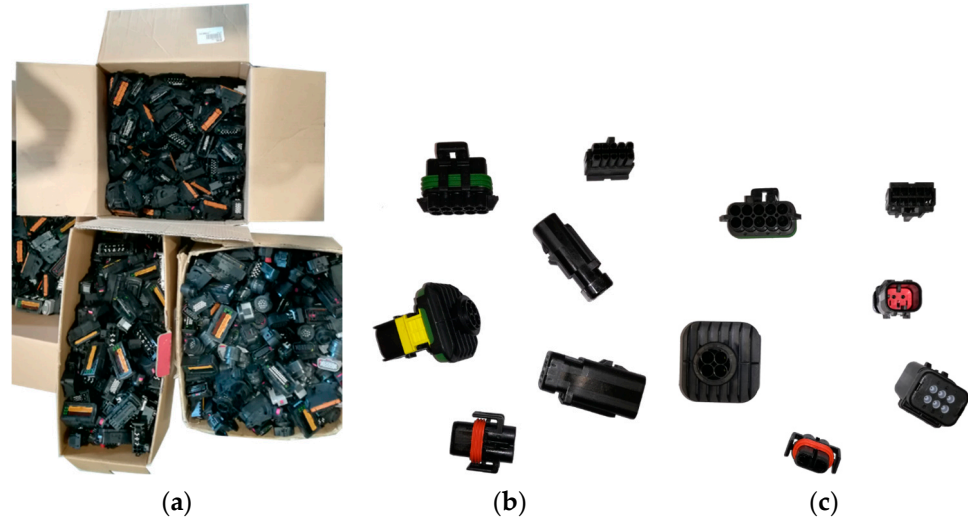


Figure 2. (a) Boxes of connectors for reuse. (b,c) Sample of some of the connectors used for recognition, in different poses.

The operation takes place in a ‘Bin-Picking Station’ (see Figure 3), which consists of a collaborative robot for parts’ manipulation, one Intel RealSense camera for stereoscopic (3D) vision, one working table, two ‘Open Trays’ containing unsorted small plastic connectors and eight ‘boxes’, where the robot will put the sorted connectors.

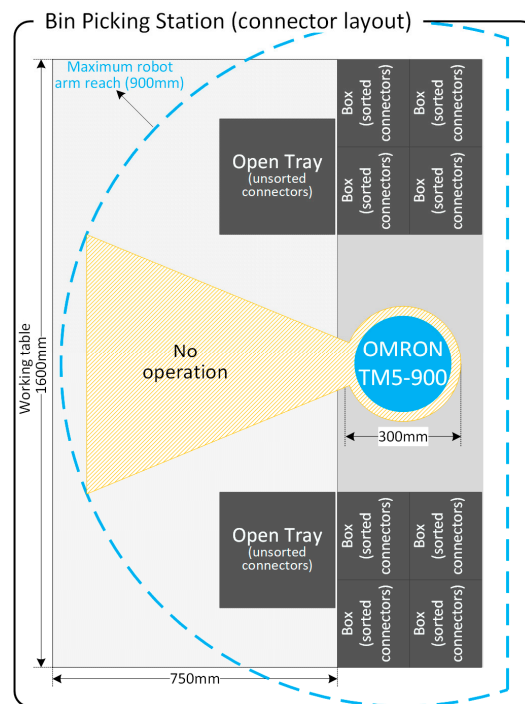


Figure 3. Layout concept for the Bin-Picking Station.

This station is responsible for grasping the connectors in the ‘Open Trays’ and sorting the connectors into the output boxes, correctly aligned to be reused in other workstations to remove seals and re-insert the connectors into the production lines. As can be seen in Figure 3, the layout has been prepared to maximize the robot’s working area and operating times. As a collaborative robot is used, it can work on one side of the station, left or right, while an operator can insert new bins and remove sorted boxes on the other side, reducing downtimes as much as possible.

The success of object detection is strongly influenced by the labeling and training of the objects to be detected. The tasks performed in the training process for any object detection are typically composed of four stages, as depicted in Figure 4. Several images of the connectors in different poses and under different lighting conditions were acquired to cover as much of the variability as possible in the Bin-Picking Workstation. Using a labeling application, by defining regions, references were created for all connectors, as depicted in Figure 5.

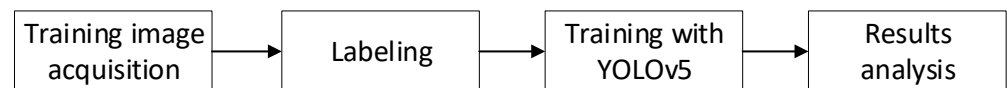


Figure 4. Typical tasks performed in the object recognition training process.

The outputs generated in the labeling application were then used in the training process. In the training task, the images and labels were organized into training, testing and validation groups. The PyTorch-based algorithm, YOLOv5, was configured and used to train the data. YOLOv5 allowed us to work with different levels of complexity associated with neural networks. At the time, four models were of interest to us: YOLOv5s (small), YOLOv5m (medium), YOLOv5l (large) and YOLOv5x (extra (X)large).

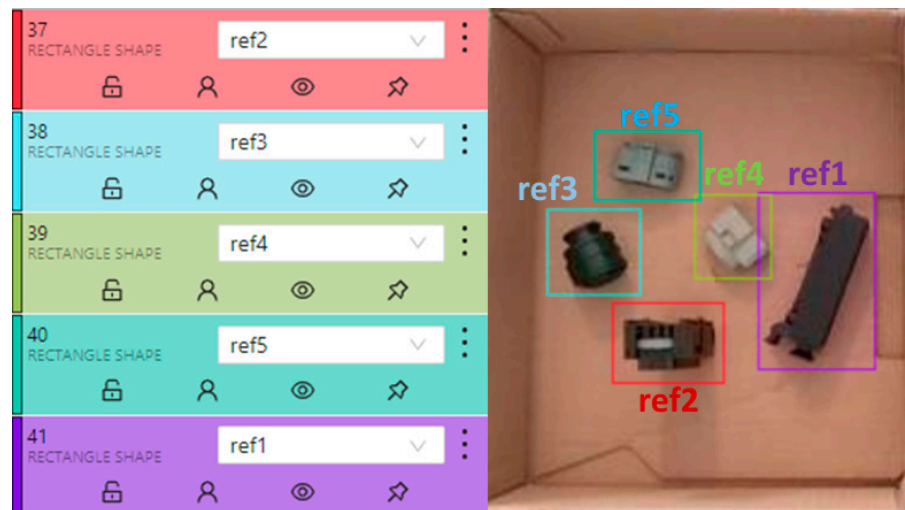


Figure 5. Creating references through a labeling application.

The main difference of each version is in the complexity and number of hidden layers of each deep neural network, which varies from simpler (small) to more complex (Xlarge). To choose the best version to use in each application, a trade-off analysis between speed, computational processing time and accuracy is required. Larger neural networks favor better accuracy results, but on the other hand the computational cost tends to be very slow, sometimes not valid for real-time applications. All YOLOv5 versions were tested and the YOLOv5s was selected for the final application, as it produced the best accuracy–speed–robustness relation for our use case. The YOLOv5 object detection algorithm works like a regression problem with three main components or sections, the Backbone, the Head and the Detection, as illustrated in Figure 6. The Backbone is a CNN that collects and models image resources at different granularities. The Head is a series of layers to combine image resources to throw them into a prediction process. Detection is a process that uses Head resources and performs box and class prediction steps. To do this, a loss function for bounding boxes' predictions based on the distance information between the predicted frame and the real frame, known as Generalized Intersection over Union (GIoU), is used. This function is proposed in [26] and described by Equation (1):

$$GIoU_{Loss} = 1 - \left(IoU - \frac{|C/A \cup B|}{|C|} \right) \quad (1)$$

where IoU is the Intersection over Union, a common evaluation metric used to measure the accuracy of an object detector, by comparing two arbitrary shapes (volumes) A and B :

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

and C is the smallest convex shape involving A and B .

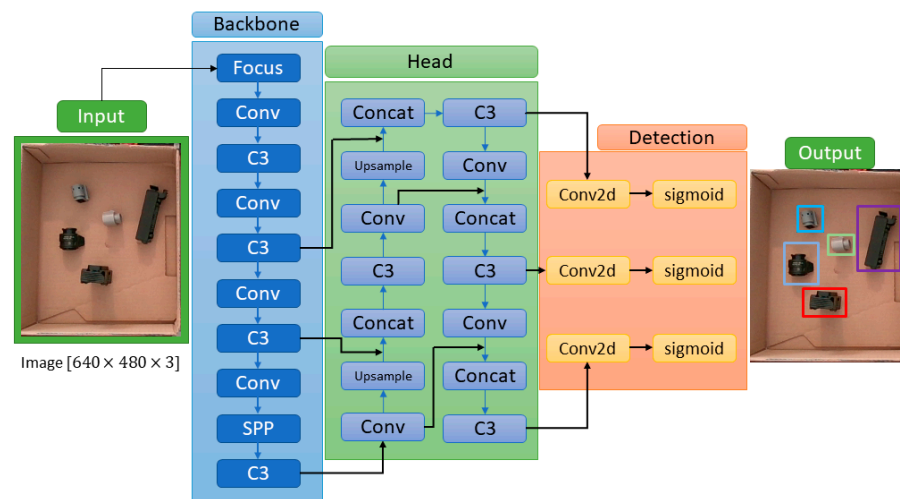


Figure 6. YOLOv5 architecture.

To obtain valid outputs, YOLOv5 requires training datasets to have a minimum of 100 images. It is known that by increasing the dataset size, the output results improve, however, there is a side effect since we need to consider the compromise between the dimension of the datasets and the associated processing time in the training process. Several training experiments were performed, with a total of about 2000 different images of connectors.

The evaluation of the classification algorithms performance is carried out by a confusion matrix. A confusion matrix is a table for summarizing the performance of a classification algorithm. Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class, or vice versa. The computation of a confusion matrix can provide a better idea of what the classification model is getting right and what types of errors it is making.

3.1. Vision-Based Robot Navigation

Whether the grasping of objects is achieved by human resources or robotic systems, it is one of the main requirements in many manufacturing processes. However, in highly dense scenes, where the environment includes several uncertainties, such as randomly distributed objects, the robot navigation is closely related to the perception of the scene. This can be affected by features of the objects of interest as well as the previous and following manipulations. Besides the picking and placing of the object, the robotic system must perceive the environment and extract the required features from it to interact with it. Precise information about the location of the object of interest in space is one of the prerequisites to obtain a proper grasp position and therefore accomplish a successful grasp [27].

The grasping process itself is usually split into several parts that typically depend on the type of gripper and the object of interest. Nevertheless, several standard grasping phases of the two-finger gripper can be identified: approaching the object, coming into contact with the object, increasing the force until the object is securely grasped and moving the object until it can be released in the place position [28]. In bin-picking settings, a surrounding-aware visual perception system in a combination with the correct approach and retreat movements is a remarkably important part of the navigation system. Objects are randomly distributed, and unintended collisions that can occur with the bin, other environmental elements and the robot itself should be strictly avoided [29].

The eye-to-hand system navigation is used to obtain each connectors' pose from the object recognition task, where the centroid of each connector is extracted and synchronized with the depth obtained by the Intel RealSense camera. To guide the robot to each position, the pose of the connector was calculated in the robot gripper reference frame.

This process is performed by a calibration procedure, as illustrated in Figure 7 and defined by Equation (3):

$${}^G T_P = ({}^B T_G)^{-1} \times {}^B T_C \times {}^C T_P \quad (3)$$

where all parameters are homogeneous matrices, with rotation and position elements.

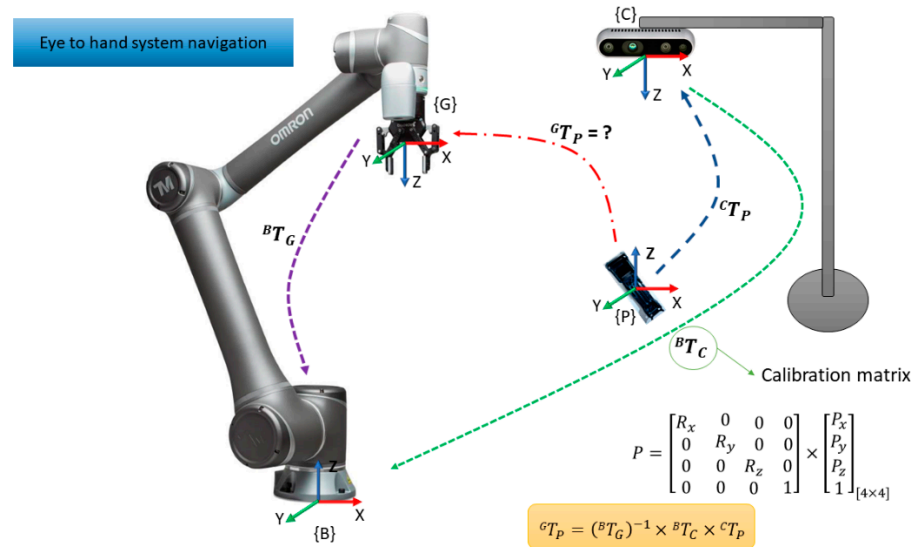


Figure 7. Coordinates reference system.

${}^G T_P$ is the homogeneous matrix that represents the pose of each recognized connector in the reference frame of the gripper. As expressed by the equation, to achieve this, it is necessary to compute the kinematics of each reference frame involved, such as the relationship between the gripper and the robot base (${}^B T_G$), the camera pose in the robot's reference frame (${}^B T_C$) and the object pose in the camera reference frame (${}^C T_P$).

This equation allows knowing, at each instant, the position and orientation of the connector to be grasped. The computation is performed in Python and the interaction with the robot controller is performed through Modbus. Trajectory planning is performed in the main program running at the controller.

4. Results

This section presents the results obtained in the training and classification tasks, as well as the solution for grasping the recognized objects. As stated in Section 3, about 2000 images were acquired in different poses and lighting conditions to obtain a large dataset that can represent the greatest possible variability of the system. With this dataset, several tests and training tasks were performed to obtain the most robust model that can be used in real-time object identification.

Table 1 presents the results of just two setups, from a larger number of setup exercises. The computation was performed by a portable computer with an Intel Core™ i7-10510U CPU running at 1.80–2.30 GHz, and with 16 GB of RAM memory.

Table 1. Time taken for training.

	Setup 1 (For Comparison)	Setup 2 (Best Results)
Image Size	640 × 480	640 × 480
Number of training images	192	552
Number of test images	40	56
Number of validation images	40	72
Method	YOLOv5s	YOLOv5s

Epochs	15	15
Batch Size	8	8
Training time	40 min	150 min

4.1. Results for Setup 1

Based on the confusion matrix depicted in Figure 8, this scenario was only able to achieve good results for the classification of ref2 and ref3, and was not able to correctly classify the references ref1, ref4 and ref5. With the weights obtained in this training, inferences were performed in real-time, and the method presents good results only for the recognition of the connectors with ref2 and ref3. This method proved to be unable to detect the other references, or confused as to how.

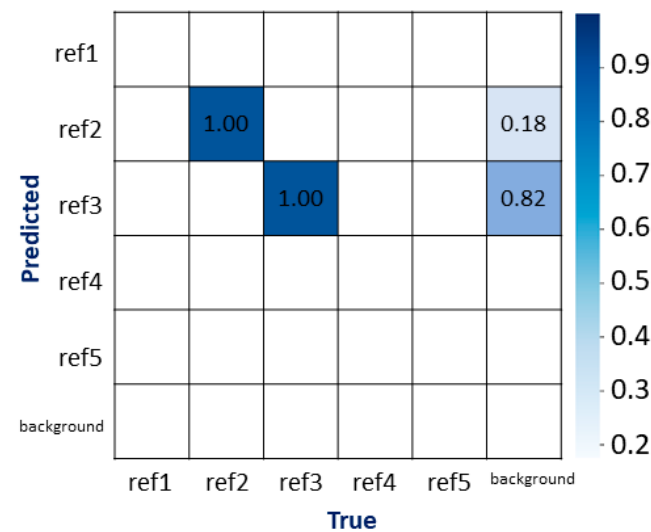


Figure 8. Obtained confusion matrix for Setup 1.

The output results for the YOLOv5s algorithm, based on the training dataset, are depicted in Figure 9. A total of 15 Epochs were used to achieve a high Precision, Recall and mean Average Precision (mAP)—additional iterations would not have led to a substantial gain. Among all the identified KPIs, these were the most relevant for our analysis, and their values were obtained using Equations (4)–(6), respectively:

$$Precision = \frac{True\ positive}{True\ positive + False\ positive} \quad (4)$$

$$Recall = \frac{True\ positive}{True\ positive + False\ negative} \quad (5)$$

$$mAP = \frac{\sum_{q=1}^Q \overline{P(q)}}{Q} \quad (6)$$

where Q is the number of queries in the dataset and $\overline{P(q)}$ is the average precision for a given query, q .

Nonetheless, by correlating the results with the confusion matrix, it can be concluded that the outcome suffers from an insufficient dataset for training (low number of samples) (only references ref2 and ref3 were correctly identified).

Figure 10 depicts classification results produced by the image testing phase of the training process. These were used to validate and fine-tune our classification model. As it can be seen, almost all connectors were classified either as ref2 or ref3. Therefore, the algorithm was not able to correctly detect the false positives.

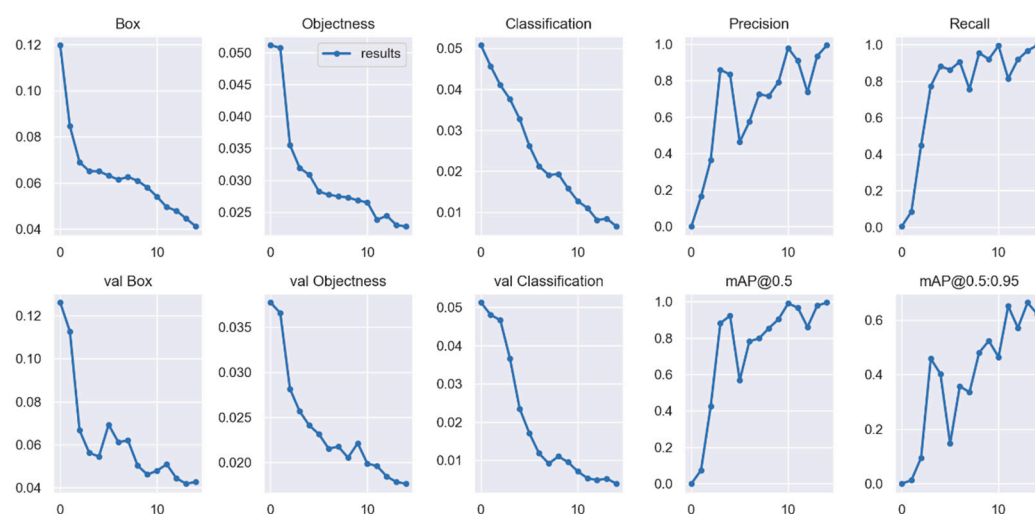


Figure 9. Setup 1 output results using the YOLO5vs algorithm.

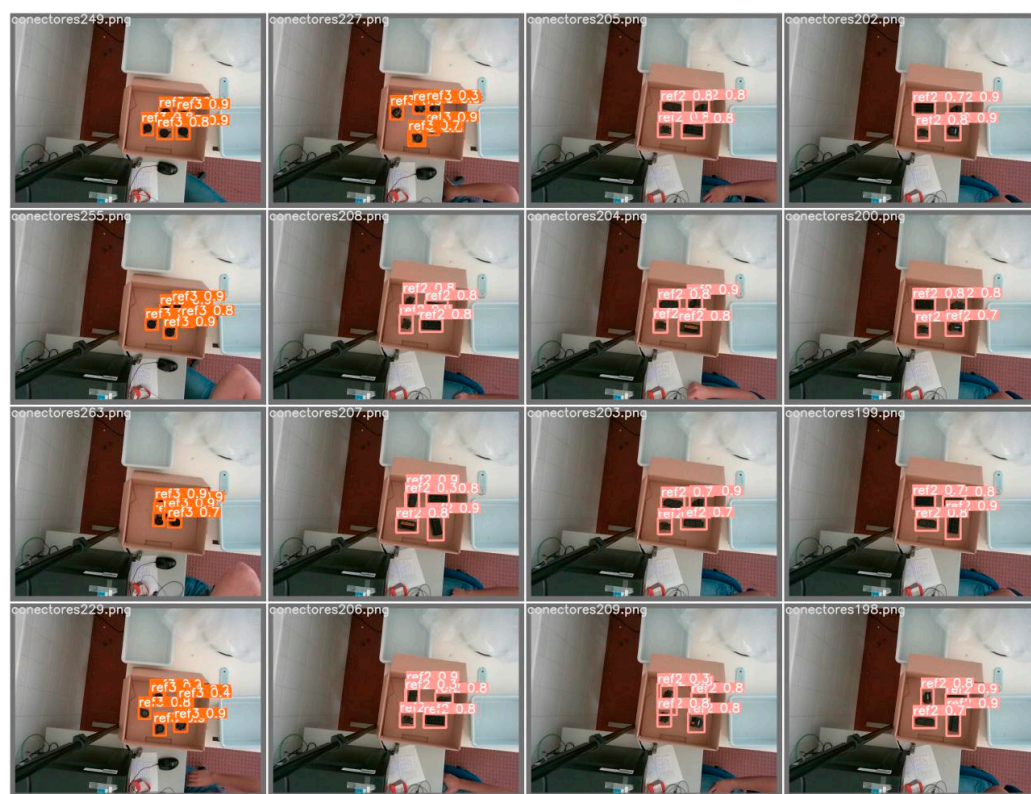


Figure 10. Setup 1 classification results using the YOLOv5s algorithm.

4.2. Results for Setup 2

Setup 2 produced better results. Despite all references being classified, due to very close similarity between ref4 and ref5, depending on their position on the tray, the system still becomes confused. These two references have the same shape and overall color—only the top layer has a different color. By correlating the output results from the confusion matrix (see Figure 11) and the results from the YOLOv5s algorithm (Figure 12), we can conclude that the results are not yet as desired, as they still suffer from an insufficient dataset for training (insufficient number of samples); however, it is now possible to classify the five references with good precision values.

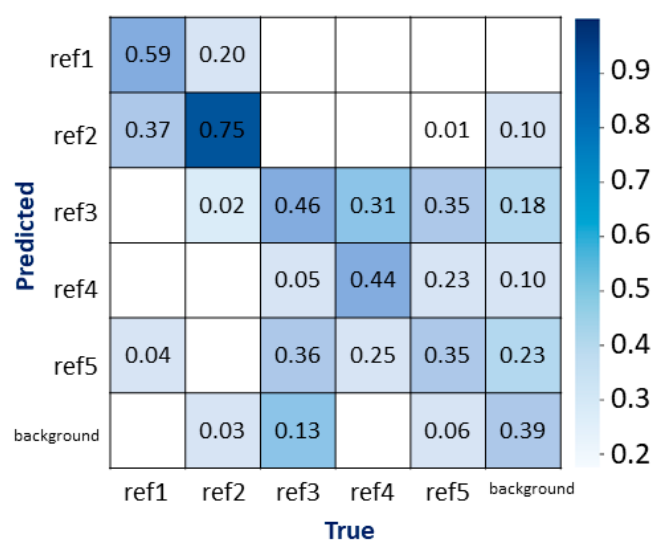


Figure 11. Confusion matrix for Setup 2.

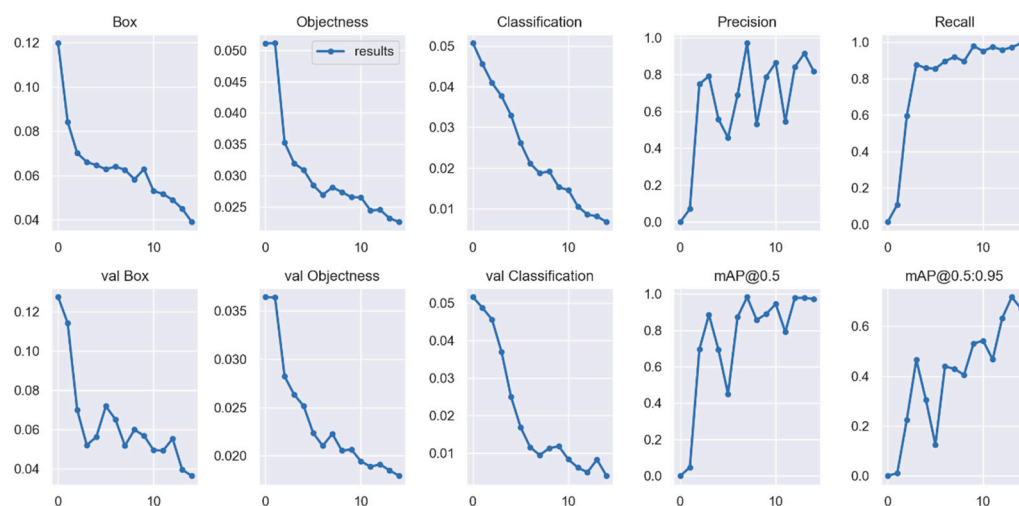


Figure 12. Setup 2 output results using the YOLOv5s algorithm.

Figure 13 depicts classification results produced by the image testing phase of the training process. These were used to validate and fine-tune our classification model. By comparing with the first experiment (Setup 1), in this experiment, all five connector references were detected with good accuracy, even in the presence of connectors with similar characteristics, such as shape and color.

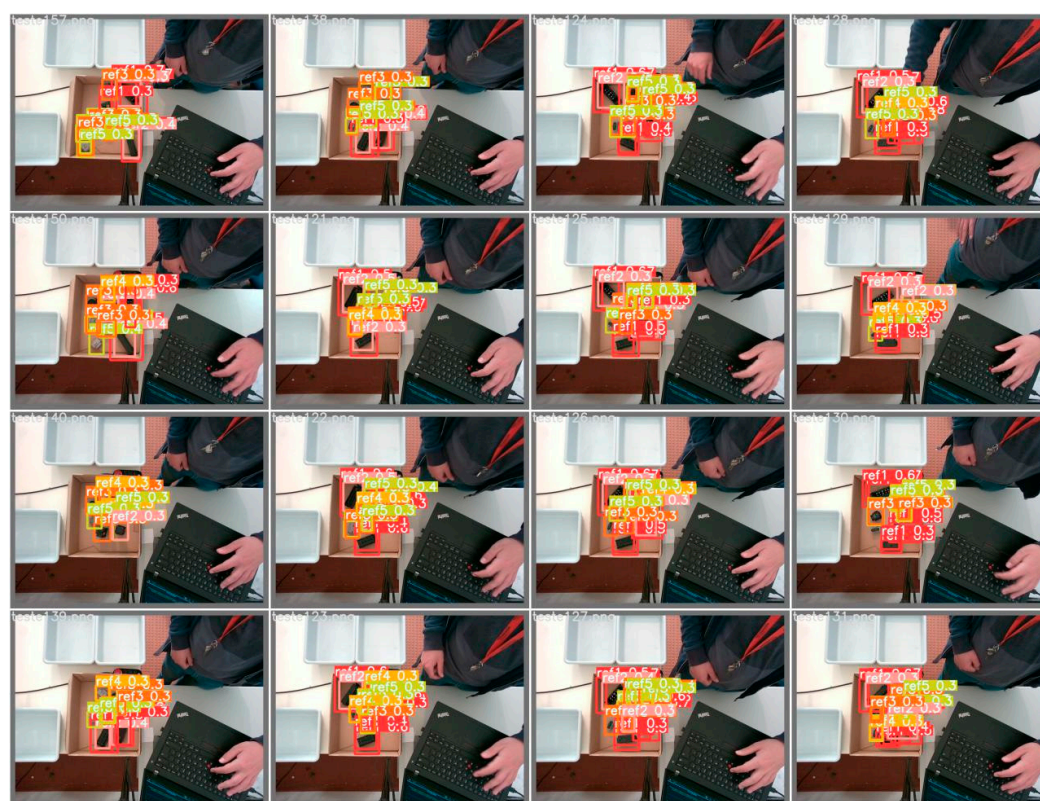


Figure 13. Setup 2 classification results using the YOLOv5s algorithm.

This trained model was the one chosen for the implementation of real-time object detection in the bin-picking solution. This choice took into account the algorithm's performance, considering its accuracy and processing time, and its comparison with other state-of-the-art algorithms, when using the same datasets. The comparison results are depicted in Table 2.

Table 2. Object detection results: comparison between different state-of-the-art algorithms.

Measure	SSD	Faster R-CNN	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
Precision	0.70	0.80	0.82	0.83	0.85	0.90
Recall	0.41	0.45	0.99	0.99	0.98	0.99
mAP@0.5	0.74	0.76	0.97	0.97	0.98	0.99
CPU time	120 min	135 min	150 min	3 h 40 min	4 h 50 min	6 h 20 min

Clearly, YOLOv5s presented better results than SSD (Single Shot Detector) [30] and Faster R-CNN [31]. We can also observe that no significant gains were obtained when using higher versions of YOLOv5, which provide similar precision but are more computationally demanding.

4.3. Identification

After the pattern recognition and pose detection, a match between the identified and trained patterns needs to take place, which occurs by moving and changing the orientation of the robot gripper, a process illustrated in Figure 14. This task has a special interest to identify the best pose for grasping each kind of object.

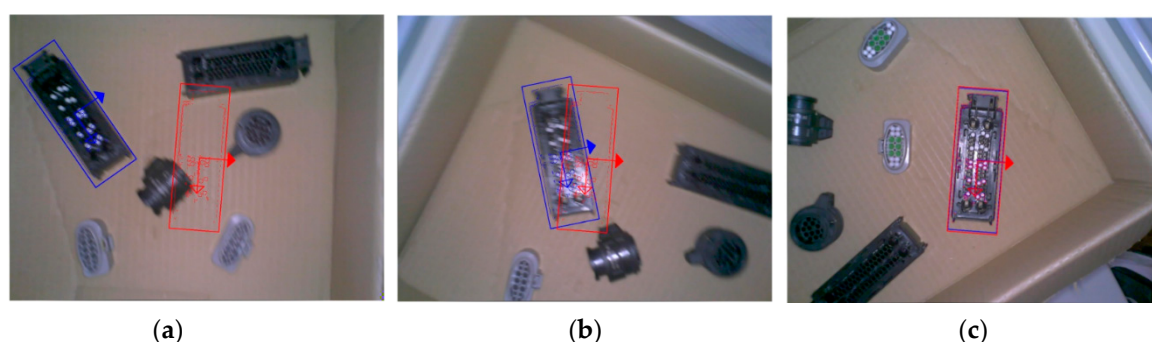


Figure 14. Steps in identifying connector orientation. (a) search for a shape and orientation. (b) approach to the shape. (c) matching shape.

The core goal was to achieve an average cycle time of 10 s for a robotic arm to successfully recognize and pick up a cable connector with unpredictable positions, thanks to AI-based machine vision. Table 3 depicts the times measured. Since this is not a collaborative operation, the average time used for the pick-and-place operation is a reference value, for non-collaborative robots.

Table 3. Average cycle time taken to successfully pick up the cable connectors with unpredictable positions.

Average time for pattern/object recognition	Average time for determining pose/orientation	Average time for connector pick-and-place in our setup (reference value for a non-collaborative SCARA robot with 3 axes)
1.83 s	5.57 s	2 s

5. Conclusions

This work aimed to demonstrate that for small objects, such as automotive connectors, bin-picking solutions with a low-cost 3D vision system are possible. The machine vision algorithm plays an important role in correctly identifying objects, and this is only possible due to the contribution of machine learning algorithms. The YOLO algorithm has been shown to have great potential for these tasks, in particular, YOLOv5 was shown to recognize these kinds of small objects with high accuracy and repeatability. Grasping this type of connector is a challenging task due to its layout not being solid and it being capable of being vacuum aspirated, which makes manipulation difficult. Our test scenario used a two-finger gripper, which implies identifying the best pose for grasping the connector, and is more propitious to collisions when interacting with very close objects. Despite these challenges, this work demonstrated that it is possible to grasp small objects in bulk, classifying them and sorting them into different output boxes.

As future work, the main focus will be further reducing the cycle time, possibly by improving the time required to identify the best posture to grip the connectors. Additionally, a new type of gripper is being considered that would be more suitable for grasping these types of objects.

Author Contributions: Conceptualization, P.T., H.M. and P.M.; methodology, P.T. and H.M.; software and hardware, P.T.; validation, P.T., J.A., H.M., and P.M.; investigation, P.T., J.A., H.M., and P.M.; writing—original draft preparation, P.T.; writing review and editing, P.T., J.A., H.M., and P.M.; funding acquisition, P.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work results from a sub-project that has indirectly received funding from the European Union's Horizon 2020 research and innovation programme via an Open Call issued and executed under project TRINITY (grant agreement No 825196).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Indicators and a Monitoring Framework. Available online: <https://indicators.report/targets/12-5/> (accessed on 26 December 2021).
2. Automotive Wiring Harness Market - Growth, Trends, COVID-19 Impact, and Forecasts (2021–2026). Available online: <https://www.mordorintelligence.com/industry-reports/automotive-wiring-harness-market> (accessed on 26 December 2021).
3. Automotive Wiring Harness Market by Application. Available online: <https://www.marketsandmarkets.com/Market-Reports/automotive-wiring-harness-market-170344950.html> (accessed on 26 December 2021).
4. Ghobakhloo, M. Industry 4.0, digitization, and opportunities for sustainability. *J. Clean. Prod.* **2020**, *252*, 119869.
5. Fujita, M.; Domae, Y.; Noda, A.; Garcia Ricardez, G.A.; Nagatani, T.; Zeng, A.; Song, S.; Rodriguez, A.; Causo, A.; Chen, I.M.; Ogasawara, T. What are the important technologies for bin picking? Technology analysis of robots in competitions based on a set of performance metrics. *Adv. Robot.* **2020**, *34*, 560–574.
6. Marvel, J.; Eastman, R.; Cheok, G.; Saidi, K.; Hong, T.; Messina, E.; Bollinger, B.; Evans, P.; Guthrie, J.; Hershberger, E.; Martinez, C. Technology Readiness Levels for Randomized Bin Picking. In Proceedings of the Workshop on Performance Metrics for Intelligent Systems, Hyattsville, MD, USA 20–22 March 2012; pp. 109–113.
7. Kleeberger, K.; Bormann, R.; Kraus, W.; Huber, M.F. A Survey on Learning-Based Robotic Grasping. *Curr. Robot. Rep.* **2020**, *1*, 239–249, <https://doi.org/10.1007/s43154-020-00021-6>.
8. Photoneo. Available online: <https://www.photoneo.com/> (accessed on 29 December 2021).
9. Zivid. Available online: <https://www.zivid.com/> (accessed on 29 December 2021).
10. Solomon. Available online: <https://www.solomon-3d.com/> (accessed on 29 December 2021).
11. Pickit. Available online: <https://www.pickit3d.com/en/> (accessed on 29 December 2021).
12. Song, Q.; Li, S.; Bai, Q.; Yang, J.; Zhang, X.; Li, Z.; Duan, Z. Object Detection Method for Grasping Robot Based on Improved YOLOv5. *Micromachines* **2021**, *12*, 1273, <https://doi.org/10.3390/mi12111273>.
13. Pochyly, A.; Kubela, T.; Singule, V.; Cihak, P. 3D Vision Systems for Industrial Bin-Picking Applications. In Proceedings of the 15th International Conference MECHATRONIKA, Prague, Czech Republic, 5–7 December 2012, pp. 1–6.
14. Martinez, C.; Chen, H. and Boca, R. Automated 3D Vision Guided Bin Picking Process for Randomly Located Industrial Parts. In Proceedings of the 2015 IEEE International Conference on Industrial Technology (ICIT), Seville, Spain, 17–19 March 2015; pp. 3172–3177, doi: 10.1109/ICIT.2015.7125566.
15. Yan, W.; Xu, Z.; Zhou, X.; Su, Q.; Li, S.; Wu, H. Fast Object Pose Estimation Using Adaptive Threshold for Bin-Picking. *IEEE Access* **2020**, *8*, 63055–63064, <https://doi.org/10.1109/access.2020.2983173>.
16. Matsumura, R.; Domae, Y.; Wan, W.; Harada, K. Learning Based Robotic Bin-picking for Potentially Tangled Objects. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 7990–7997, doi: 10.1109/IROS40897.2019.8968295.
17. Lee, S.; Lee, Y. Real-Time Industrial Bin-Picking with a Hybrid Deep Learning-Engineering Approach. In Proceedings of the 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), Busan, Korea, 19–22 February 2020; pp. 584–588, doi: 10.1109/BigComp48618.2020.00015.
18. Arents, J.; Cacurs, R.; Greitans, M. Integration of Computervision and Artificial Intelligence Subsystems with Robot Operating System Based Motion Planning for Industrial Robots. *Autom. Control Comput. Sci.* **2018**, *52*, 392–401, <https://doi.org/10.3103/s0146411618050024>.
19. Kim, K.; Kang, S.; Kim, J.; Lee, J.; Kim, J. Bin Picking Method Using Multiple Local Features. In Proceedings of the 2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Goyang-si, Korea, 28–30 October 2015; pp. 148–150, doi: 10.1109/URAI.2015.7358848.
20. Pyo, J.; Cho, J.; Kang, S. and Kim, K. Precise Pose Estimation Using Landmark Feature Extraction and Blob Analysis for Bin Picking. In Proceedings of the 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Jeju, Korea, 28 June–1 July 2017; pp. 494–496, doi: 10.1109/URAI.2017.7992786.
21. Dolezel, P.; Pidanic, J.; Zalabsky, T.; Dvorak, M. Bin Picking Success Rate Depending on Sensor Sensitivity. In Proceedings of the 2019 20th International Carpathian Control Conference (ICCC), Krakow-Wieliczka, Poland, 26–29 May 2019, pp. 1–6, doi: 10.1109/CarpathianCC.2019.8766009.
22. Zhao, Z.-Q.; Zheng, P.; Xu, S.-T.; Wu, X. Object Detection With Deep Learning: A Review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232, <https://doi.org/10.1109/tnnls.2018.2876865>.
23. Thuan, Docong. “Evolution Of Yolo Algorithm And Yolov5: The State-Of-The-Art Object Detection Algorithm.” (2021).
24. Xu, Q.; Zhu, Z.; Ge, H.; Zhang, Z.; Zang, X. Effective Face Detector Based on YOLOv5 and Superresolution Reconstruction. *Comput. Math. Methods Med.* **2021**, *2021*, 1–9, <https://doi.org/10.1155/2021/7748350>.
25. Yao, J.; Qi, J.; Zhang, J.; Shao, H.; Yang, J.; Li, X. A Real-Time Detection Algorithm for Kiwifruit Defects Based on YOLOv5. *Electronics* **2021**, *10*, 1711, <https://doi.org/10.3390/electronics10141711>.

26. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and A Loss for Bounding Box Regression. In Proceedings of the IEEE/CVF Conference On Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 658–666.
27. Zhang, B.; Xie, Y.; Zhou, J.; Wang, K.; Zhang, Z. State-of-the-art robotic grippers, grasping and control strategies, as well as their applications in agricultural robots: A review. *Comput. Electron. Agric.* **2020**, *177*, 105694, <https://doi.org/10.1016/j.compag.2020.105694>.
28. Fantoni, G.; Santochi, M.; Dini, G.; Tracht, K.; Scholz-Reiter, B.; Fleischer, J.; Lien, T.K.; Seliger, G.; Reinhart, G.; Franke, J.; et al. Grasping devices and methods in automated production processes. *CIRP Ann.* **2014**, *63*, 679–701, <https://doi.org/10.1016/j.cirp.2014.05.006>.
29. Guo, J.; Fu, L.; Jia, M.; Wang, K.; Liu, S. Fast and Robust Bin-picking System for Densely Piled Industrial Objects. **2020**, 2845–2850, <https://doi.org/10.1109/cac51589.2020.9327193>.
30. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single Shot Multibox Detector. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
31. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99.