applied sciences

MDPI

Article

Development of an Image Processing Application for Element Detection in a Printed Circuit Board Manufacturing Cell

Antonio Trejo-Morales *, Milton Bautista-Ortega, Leonardo Barriga-Rodríguez, Celso Eduardo Cruz-González and Edgar Adrián Franco-Urquiza

Ingeniería de manufactura, Centro de Ingeniería y Desarrollo Industrial, Av. Playa Pie de la Cuesta No. 702, Desarrollo San Pablo, Santiago de Querétaro 76125, Querétaro, Mexico; milton.bo343@gmail.com (M.B.-O.); lbarriga@cidesi.edu.mx (L.B.-R.); ecruz@cidesi.edu.mx (C.E.C.-G.); edgar.franco@cidesi.edu.mx (E.A.F.-U.)
* Correspondence: atrejo@cidesi.edu.mx; Tel.: +52-442-211-9800

Abstract: Industrial automation in the manufacturing environment has revolutionized production and manufacturing in many industries, generating significant improvements in efficiency, quality, and process effectiveness. However, it has also posed challenges related to feedback in manufacturing environment monitoring systems, and increasing the effectiveness, productivity, and quality in industrial production. Feedback systems in the manufacturing environment are fundamental to industrial automation, which is why an application has been developed for the detection of elements in a printed circuit board manufacturing cell. The solution presented in this article proposes implementing a continuous feedback system with the ability to provide real-time information to identify the location of elements in a manufacturing cell and potentially detect anomalies, with the goal of improving the manufacturing process appropriately.

Keywords: anomaly identification; feedback systems; industrial automation; manufacturing cell; manufacturing process

Citation: Trejo-Morales, A.; Bautista-Ortega, M.; Barriga-Rodríguez, L.; Cruz-González, C.E.; Franco-Urquiza, E.A. Development of an Image Processing Application for Element Detection in a Printed Circuit Board Manufacturing Cell. *Appl. Sci.* **2024**, *14*, 5679. https://doi.org/10.3390/app14135679

Academic Editor: Guijun Bi

Received: 20 May 2024
Revised: 20 June 2024
Accepted: 25 June 2024
Published: 28 June 2024

1. Introduction

The Fourth Industrial Revolution (4IR) was coined by the World Economic Forum in 2016 in order to raise global income levels, thus improving life quality for the worldwide population [1]. In a strict sense, there are still remaining doubts about what exactly the 4IR means in terms of scope and impact in technology development. The first industrial revolution (1784) yielded the use of steam to move mechanical equipment. Then, the second (1870) and third ones (1969) developed work division/mass production and automated production, respectively. Finally, the named 4IR was originated based on internet employment and connectivity [2,3]. However, COVID-19 opened a whole spectrum of opportunities ranging from remote medical consulting to digital manufacturing (DM).

The global printed circuit board (PCB) manufacturing industry experienced a reduction in 2023 due the imbalances that resulted from the COVID-19 pandemic, and will currently grow by up to about 6.3%. In addition, a reported prediction suggests a market growth of about 12.5% for Flexi boards around 2031 [4]. Since China is the main PCB producer, due to the abundance of raw materials and advanced manufacturing technology, it points to opportunities to develop competitive technology for this market [5]. In that sense, the planning of this kind of project requires modern techniques that could help to evaluate them at early stages of new technology adaptation, or in terms of productivity. This suggests that the new paradigm, the DM, can be useful to help investors, researchers, and students to make better decisions; some examples are in references [6–8].

The DM involves several concepts and fancy automation techniques. One such technique is artificial intelligence (AI), which permits computers to simulate human behavior (intelligence and capacity) [9,10]. An interesting branch of AI is computer vision (CV), because it is able to provide recommendations, which distinguishes it from image recognition

tasks [9]. Thus, providing crucial information thorough detailed image analysis for the decision-making process within manufacturing operations is the cornerstone of the CV technique. In addition, the employment of sophisticated algorithms will enable the system to learn and optimize operations related to decision making.

Image analysis through algorithms was employed by Yasnyy-Boyko [11] to process images within a reconfigurable device. The above was attained by means of the Xilinx and Vivado tools, then the HW was designed through the $C/C++$ language. However, due technical limitations in the development plate, it will be changed to obtain better results. In contrast, Reis et al. [12] employed the Yolov8 model to detect 40 different classes of flying objects, and their results suggests that model mean average precision reached 99%. Thus, the authors obtained results within the state of the art.

Bazame et al. [13] employed CV and an object detection system (YOLOv3-tiny) to detect, classify, and map coffee fruits during their harvesting. In this work, the authors recorded 90 videos at the discharge conveyor of arabica coffee (Catuaí 144) on a Brazilian farm during 2020. The model peaked at the 3300th iteration when an 800 × 800 pixel image resolution was reached. Their results reached precisions of about 86%, 85%, and 80% for unripe, ripe, and overripe coffee fruits, respectively. Thus, this work could incentivize studies into the precision agriculture field and attributive mapping as well. A similar work was published by Chang and Huang [14], reaching efficiencies of around 95.2% after they analyzed 452 images on average per sample type.

Defect detection through the Yolov8 algorithm and CV in manufacturing was employed by Liu and co-workers [15]. The above was a complementary technique to automatic detection methods such as eddy current, infrared, and magnetic flux leakage, since such techniques have some environmental limitations. Their results depict a generalized enhancement in the operation, since the precision increases from 98.1% to 99.3%. Another work related to CV for manufacturing processes is presented in reference [16], where promising results open a wide range of possibilities for DM.

Abdullah and co-workers [17] implemented a CV-based robotic arm for object color, shape, and size detection, by using the image analysis procedure with the PixyCMU camera sensor to distinguish multiple objects according to their colors, which, in their work, were red, yellow, and green. After 10 repetitions, an average accuracy of 80% was obtained, with the red-colored one having the highest accuracy (90%) when compared to the other two. However, the accuracy of the color detection was influenced by changes in lighting conditions and the object distance. Finally, the authors reported an accuracy of 100% for shape detection characteristics (circle, triangle, and rectangle, to mention few).

Du et al. [18], published a work related to stereo vision-based object recognition by means of convolutional neural network (R-CNN). Their results suggest that R-CNN was able to recognize triangle blocks with 100% for accuracy, recall, and precision. On the other hand, a precision of about 96.88% in identifying square blocks was obtained, while 100% and 98.44% for recall and accuracy were obtained, respectively. Similarly, Rentería-Vidales [19] reported the application of a CNN for stereo vision. In that sense, the authors proposed the new ModuleNet model for stereoscopic vision, showing that, both qualitatively and quantitatively, they performed the Census–Hamming approach.

The manufacturing processes yield challenges when the operations do not obtain the planned results due to unforeseen factors. In that sense, Son et al. [20] proposed a digital twin (DT)-based cyber-physical system for automotive body production lines. The DT developed was designed to predict if a certain model of automotive body can be manufactured within a required schedule for a customer by considering abnormal scenario occurrence. The results suggest that an average prediction performance of 96.83% can be obtained for the actual production plan that includes the product, process, plan, plant, and resource information model. Previously, Icarte-Ahumada [21] proposed a DT for the supply chain in different scenarios.

As mentioned before, the 4IR relies on DT for the development of the DM, which can help to prevent unforeseen events. Also, DT can be usefully employed to scale plan

manufacturing activities that can be extrapolated to a productive process. In that sense, the aim of this work is to present a scaled manufacturing system for PCBs that includes CV technology to process images, thus achieving process segmentation in images of a PBC scaled manufacturing system. The plan for the above is to build a DM scaled system that can be used to predict, plan, and actualize a complete manufacturing plant for the components mentioned before.

Different types of manufacturing cells, especially those involving human interaction for tasks such as adding or removing supplies, adjusting elements, or simply interacting with the cell, can experience changes in element positions, omissions of necessary components, or the introduction of foreign objects. These variations can adversely affect the proper functioning of the cell, particularly the robotic arms responsible for transporting boards between machines, inspection points, and cleaning stations in the case study. Implementing an image processing system that identifies elements within the manufacturing cell provides advance information on the presence and position of essential components for proper operation. This enables the adjustment of processes executed by the manufacturing cell, particularly the actions of the robotic arms. For example, if an element required for a preprogrammed action is missing, the system can omit that action and perform only those tasks that are feasible based on the detected elements. Similarly, if a variation in the position or orientation of a component is detected, the system will adjust the robot's movements to perform the task correctly. This approach is not only applicable to PCB manufacturing cells but also to any type of cell requiring flexibility in response to variability in the conditions of its elements.

This article is divided into several sections. In Section 2, we describe the materials and methods used for element detection in a printed circuit board manufacturing cell. In Section 3, we present the results. Section 4 discusses element detection in a printed circuit board manufacturing cell. In Section 5, we present the conclusion. Finally, we list the references that support this work.

2. Materials and Methods

The detection and classification of elements in the manufacturing cell of printed circuit boards require an artificial vision application that consists of identifying the elements found by cameras that are in constant continuous feedback at runtime, sending the image processing information captured by the machine vision system to the continuous feedback system, which will be giving an interpretation of real-world objects to the observer who requires additional control measures for the effective operation of the system. The following materials are used (see Table 1) for the process described in Figure 1. The general approach to neural network implementation is using YOLOv8 as the training model and conducting tests through the Postman API.

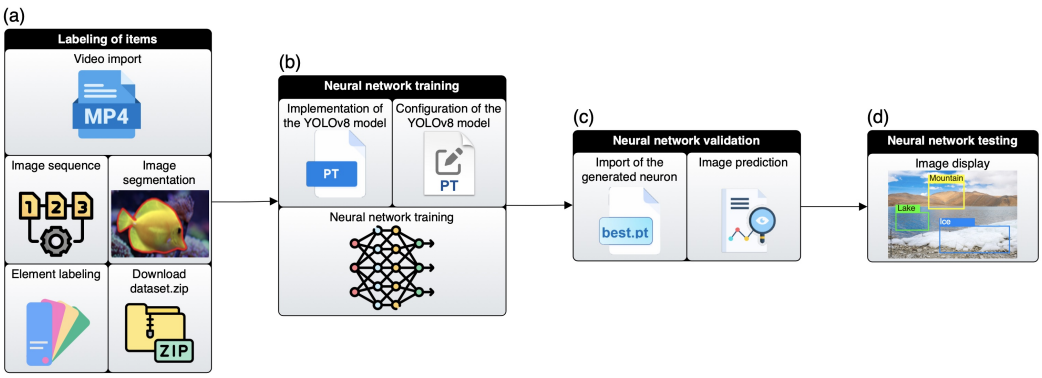


Figure 1. General proposal for the implementation of the neural network. In (**a**), a precise system of labeling the input items is established. In (**b**), a workflow is designed to train the network with labeled data. In (**c**), once the network has been trained with YOLOv8, it is validated. Finally, in (**d**), extensive testing is performed on the neural network.

Executing the overall approach at each step may require considerable time and complexity. And is summarized in the following four essential steps: labeling of items, training, validation, and testing of the neural network:

1. Labeling of items: In this initial phase, a precise system of labeling the input items is established. A coherent and detailed label structure is designed to accurately identify and classify the objects of interest in the input data.
2. Neural network training: The neural network architecture is based on the YOLOv8 model, known for its high efficiency in detecting objects in images and videos. The YOLOv8 architecture is configured and a workflow is designed to train the network with labeled data. During this process, the weights and parameters of the network are adjusted to optimize its detection and classification capabilities.
3. Neural network validation: Once the network has been trained with YOLOv8, it is validated. A validation system is designed that includes specific evaluation metrics to measure the performance of the network in object detection. The network is verified to meet the expected accuracy and performance specifications.
4. Neural network testing: Finally, extensive testing is performed using the Postman API. Test stages are designed to simulate various real-world conditions. The Postman API will facilitate test automation and the generation of detailed reports on the performance of the network in different situations.

Table 1. Description of materials used.

Name	Description	Value
Dataset: study zone	PCB manufacturing cell, CIDESI, Querétaro, Qro., México	900 frames
RoboFlow [22]	The image labeling platform requires a dataset containing test, training, and validation images. And a list of tags, Roboflow Inc., Des Moines, Iowa, EE.UU	dataset.zip, 2024 version
Python [23]	Programming language for the validation of neural network, Python Software Foundation, Wilmington, Delaware, EE.UU	3.8 version
Google Colab [24]	Develop the program in Python language for training and validation of neural networks, Mountain View, CA, EE.UU	2024 version
Ultralytics [25]	Build, train, and deploy computer vision and deep learning models using the YOLOv8 model, Ultralytics Inc., Madrid, Spain	8.0.173 version
YOLOv8 [26]	Run-time object detection and segmentation model using image processing and computer vision, Ultralytics Inc., Madrid, Spain	n-seg version
OpenCV [27]	Open-source computer vision library using image processing and computer vision, Intel Corporation, Santa Clara, California, EE.UU	4.8.0.74 version
Lambda server [28]	Workstation to process, store, and manipulate the data that will lead to the creation of the neural network, with NVIDIA GPU, CUDA, Tensor, Ubuntu 22.04.4 LTS S.O, Lambda Inc., San Jose, CA, EE.UU	RTX4090, 16,384 CUDA
Google Chrome [29]	Web browser developed by Google used to visualize the resulting image, Mountain View, CA, EE.UU	125.0.x version
Visual Studio Code [30]	Free version, integrated development environment for software creation, Redmond, Washington, EE.UU	1.89.1 version

The application of this machine vision system is divided into three zones of the manufacturing cells (zones "A, B, and C"), and each zone is monitored by a camera with the machine vision system (see Figure 2). That will allow us to identify and classify the elements found by the system according to the images that were used during pre-processing of images for the detection of the elements.

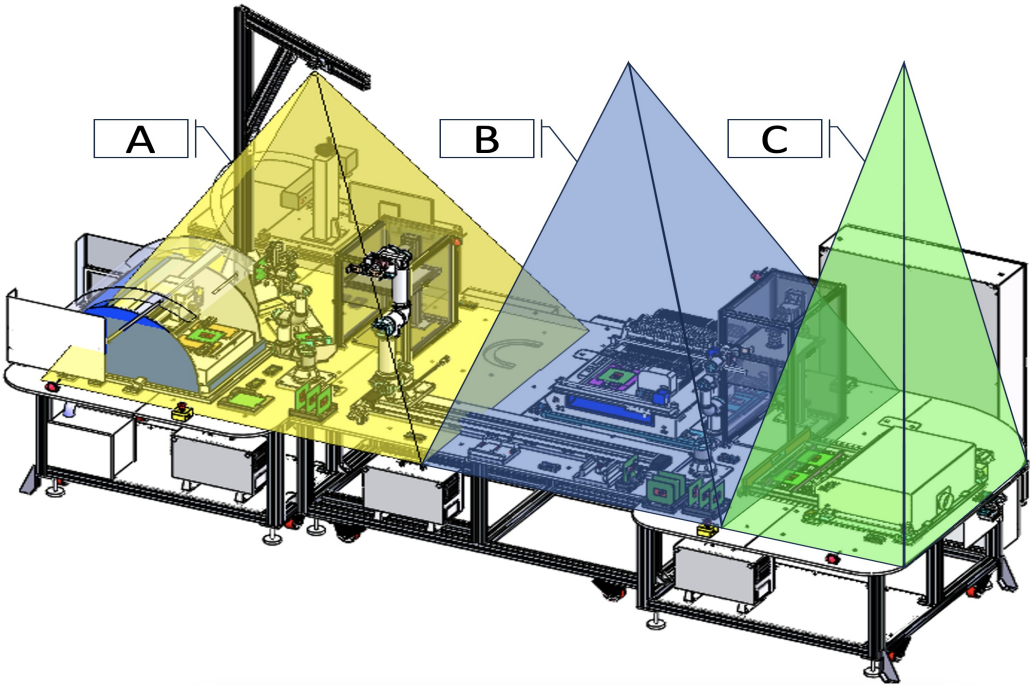


Figure 2. Overview of the manufacturing zones. (**A**) Illustrates inspection "A" zone. (**B**) Illustrates inspection "B" zone. (**C**) Illustrates inspection "C" zone. From these zones, images are obtained to train, test, and verify the effectiveness and accuracy of the implemented YOLOv8 model.

In addition, for each area of the PCB manufacturing process, a model was trained for detecting the elements that needed to be identified, so to validate the efficiency and accuracy of image processing for each zone, tests were conducted to verify the effectiveness of the implemented YOLOv8 model in computer vision of the manufacturing environment. Next, in Figure 3, the industrial design of the "A" zone by which the manufacturing cell is formed is best presented, where tests will be carried out in a simulation environment that faithfully reproduces the conditions and variables of the actual environment of the manufacturing cell. This ranges from machine layout to material flow, cycle times, and other key parameters. The simulated stage will be essential to evaluate and improve production processes, enabling more accurate and efficient decision making in manufacturing management.

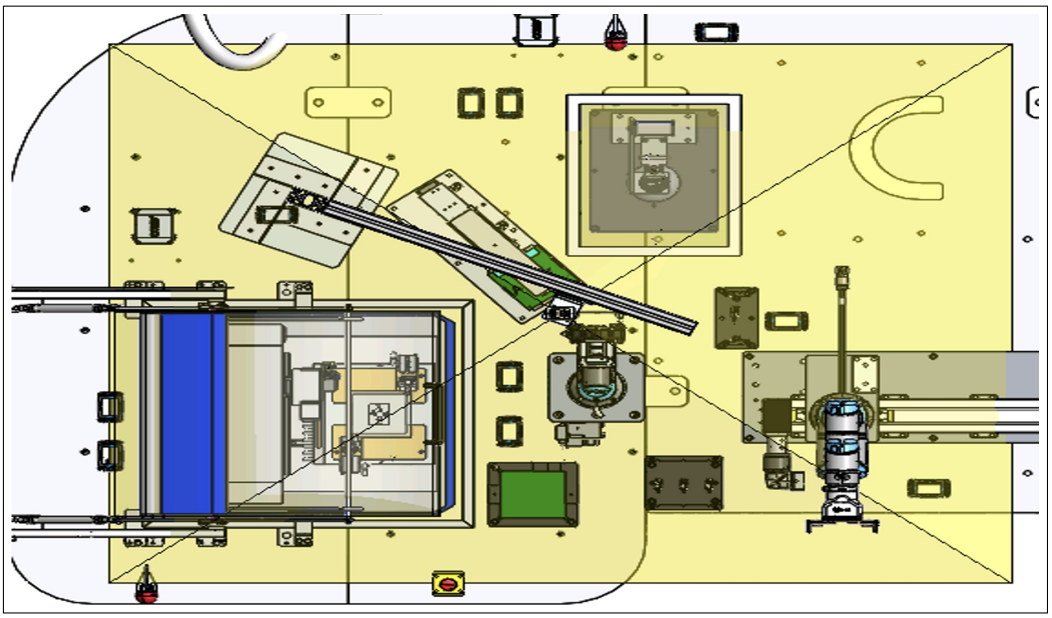


Figure 3. View of zone "A" of the manufacturing cell.

2.1. Labeling of Items

The first stage begins with an import of the video of the PCB manufacturing cell, preferably in (*.mp4) format, which will be processed through the Roboflow platform (image tagging platform) to obtain a sequence of images of the frames (a concrete image within a sequence of moving images) that contains the video. To achieve element tagging, the number of classes and labels for each identified element in every stage was implemented, as illustrated in Figure 4. This figure includes annotations of the detected elements that were incorporated into the dataset. The quantity of labels or classes required for annotation varies with each stage because it represents the actual objects or elements present in the physical space. Consequently, the number of classes per element can fluctuate depending on the specific zone assigned to each stage. For every stage, we utilized 300 images to train the dataset, as indicated in the top right corner of the image. The dataset consisted of 300 images per stage or zone in the printed circuit board manufacturing cell (900 images by 3 zones). Upon creation, the dataset is divided into folders (210 images for training, 60 images for validation, and 30 images for testing).

During dataset download, images were automatically sorted into separate folders with respective configurations (training, validation, and test), ensuring that each image had dimensions of 640 × 640 pixels and a resolution of 96 dpi. The image segmentation process will be carried out individually for each image and each element identified in the manufacturing area. For this purpose, the toolbox on the right side of the image is used (see Figure 4). We use the "Smart Polygon (S)" tool to automatically outline each selected element. Each element will be segmented and colored differently to identify it.

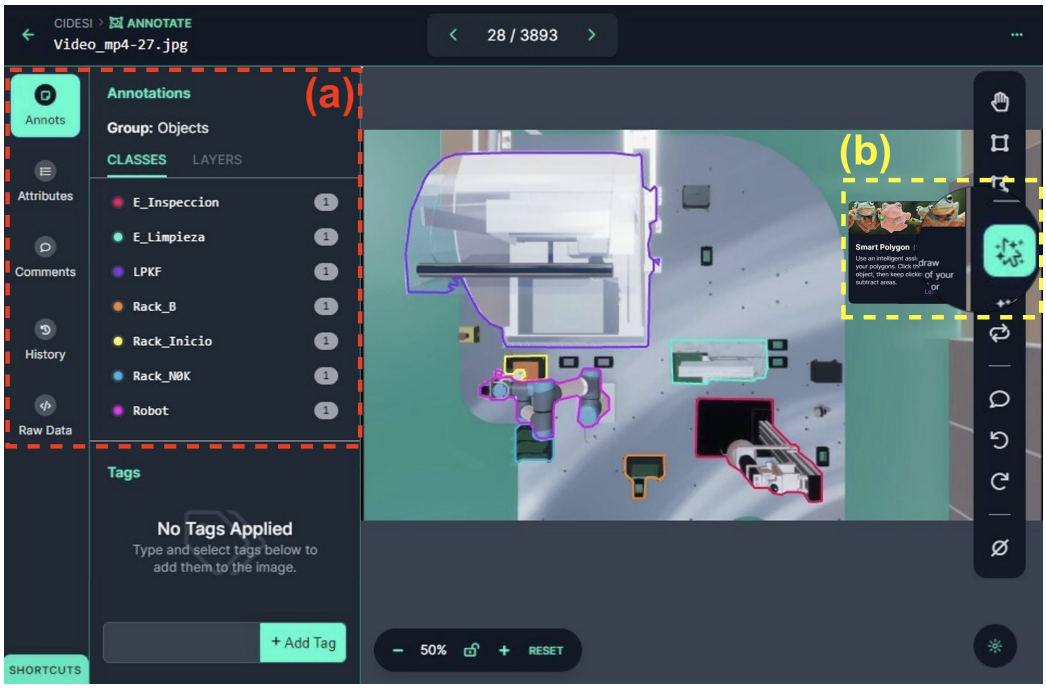


Figure 4. Segmentation and labeling of elements in the image of the "A" zone. (**a**) Illustrates the group of objects labeled with their assigned name. (**b**) Illustrates the "Smart Polygon (S)" tool to automatically outline each selected element.

Additionally, manual refinement of each element's segmentation will be possible using the polygon tool. This tool allows for the precise shaping of manufacturing cell elements that require more accurate segmentation to enhance their identification in artificial intelligence image processing. Each segmented element will be assigned its detected name in the annotation editor (see Figure 4) and will be saved within a group of objects labeled with their assigned name. Furthermore, in the left panel within the class session, labels for each detected element in the image will be added to a list. The number displayed next to

each label indicates the number of detected elements of the same class or label for a given object. Finally, the dataset is downloaded as a (*.zip) file and will be used in the second stage. The dataset contains the labeled images and a configuration file to be used in the training of the neural network.

The instance segmentation technique is effective for the application because you can know exactly where the object to be recognized is located and, as a result, can almost always identify the same object, regardless of the position in which it is located.

2.2. Neural Network Training

The second stage begins with the implementation and configuration of parameters in the YOLOv8 (You Only Look Once version 8) neural model, which was downloaded from the official Ultralytics website for inclusion in the neural network training. It offers essentially two types of image recognition techniques: image segmentation and object detection. The model introduces new features comparable to the previous models, like YOLOv5 (You Only Look Once version 5), which had good performance and improved hyperparameter optimization to set up the neural network, but instead, the YOLOv8 model introduced new features to the model, which also improves the performance, flexibility, and efficiency in computer vision AI (artificial intelligence) tasks. The purpose of using this model implies that individual objects can be precisely and accurately identified in an image and segmented from the rest of the image to outline or contour the image appropriately to its shape and then rigorously match the contour of the identified object.

Run instance segmentation with the Ultralytics YOLOv8 model (software specialized in computer vision) requires training in Python (programming language) to be able to generate an artificial neuron which executes the predictions in the images of the manufacturing cell of PCBs, in order to be able to make inferences about the results that we obtain through the system of artificial vision that will be collecting the information about the manufacturing process of the printed circuit boards.

We conducted multiple neural network training sessions with different YOLOv8 models and found that the YOLOv8n–seg [31,32] model produced the best results (see Table 2). Compared to other pre-trained models used for instance segmentation, YOLOv8n–seg excelled in identifying the locations and exact shapes of objects in an image.

Table 2. List of pre-trained models YOLOv8x-seg.

Model	Size (Pixels)	$mAP^{box\ 50:95}$	$mAP^{mask\ 50:95}$	$Speed^{CPU\ ONNX}$ (ms)	$Speed^{A100\ TensorRT}$ (ms)	Params (M)	FLOPs (B)
YOLOv8n–seg *	640	36.7	30.5	96.10	1.21	3.40	12.6
YOLOv8s–seg	640	44.6	36.8	155.7	1.47	11.8	42.60
YOLOv8m–seg	640	49.9	40.8	317.0	2.18	27.3	110.2
YOLOv8l–seg	640	52.3	42.6	572.4	2.79	46.0	220.5
YOLOv8x–seg	640	53.4	43.4	712.1	4.02	71.8	344.1

* See: https://docs.ultralytics.com/tasks/segment (accessed on 26 June 2024).

The configuration of the model is implemented using a Python script made from the Google Colab platform (platform for developing Python code from the web browser) to which the OpenCV libraries were imported (Artificial Intelligence Library). For image processing and computer vision of the manufacturing cell, hyperparameters were configured (adjusting weights and network connections), with which the neural network will be trained in such a way that the training has a high rate of effectiveness.

The parameters considered in the configuration of the YOLOv8n–seg model are listed below:

1. Lr0: The initial learning rate for the optimizer.
2. Adj. Factor: Parameter used to adjust the model.
3. Total Img: The total number of images used in the dataset.
4. Imgsz: The size of the images used for training and testing.
5. Data: The file path to the YAML file that contains dataset information.

6. Batch size: The number of images processed together in one iteration.
7. Epochs: The number of epochs and complete passes through the entire training dataset.
8. Optimizer: The optimization algorithm used to minimize the loss function and update the model parameters.
9. Conf: The minimum confidence threshold score required for a detection to be considered valid.

These parameters determine how the model learns from the data, optimizing the accuracy of the item detection system and ensuring efficient and adaptive operation in the complex manufacturing environment. The goal of this step in training the neural network is for the application to receive an image via an HTTP request using the Postman API. This image serves as input for the manufacturing cell element detection system. The system processes the image and detects objects, providing detailed information about the detected objects, such as the location of their pixels. It uses a standard resolution of 96 pixels per inch (DPI) to convert pixel coordinates to centimeters. Each detected object undergoes processing, ensuring necessary conversions to store coordinates in centimeters. Information about the detected objects is presented in JSON format, including their coordinates in both pixels and centimeters, as well as the image dimensions in both units. These results are displayed on a web interface developed to visualize images from the PCB manufacturing cell in different zones. This approach enables detailed and accurate analysis of the detected objects, facilitating control and monitoring of the manufacturing process through the model trained with the neural network.

2.3. Neural Network Validation

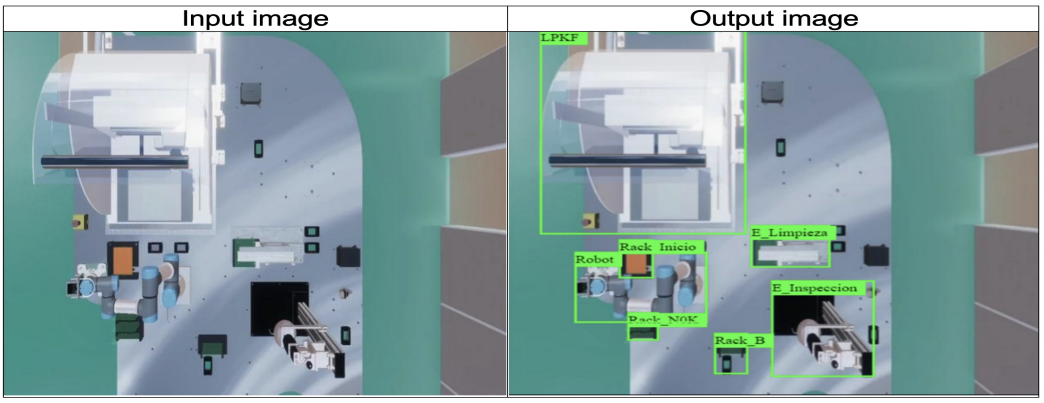
The third stage of validation of the neural network can take a long time due to the resources of the available computer equipment. The training time for the neural network model ranges from approximately 4 to 8 min, provided a GPU is used to expedite the process. Otherwise, training the neural network could take anywhere from 4 to 8 h. At the end of the training, a neuron is automatically generated, which is a file with the extension (*.pt); generally the file is called best.pt, and it is the one that contains the information of the characteristics, positions, and names of the elements found in the sequence of images that were analyzed. Subsequently, the file has to be imported into the script programmed in Python language to predict an image and identify the elements of the manufacturing cell. Using an Application Programming Interface, the image is visualized from the Google Chrome web browser [33]. Neural network validation helps identify whether the network is processing or generalizing well from the training data or if the neural network is auto-adjusting. If the network is learning too much about training data and ignoring general patterns of neural network connections, the hyperparameters of the neural network would have to be re-adjusted again so that the network predictions improve and the data are well generalized to obtain a better neural network model.

2.4. Neural Network Testing

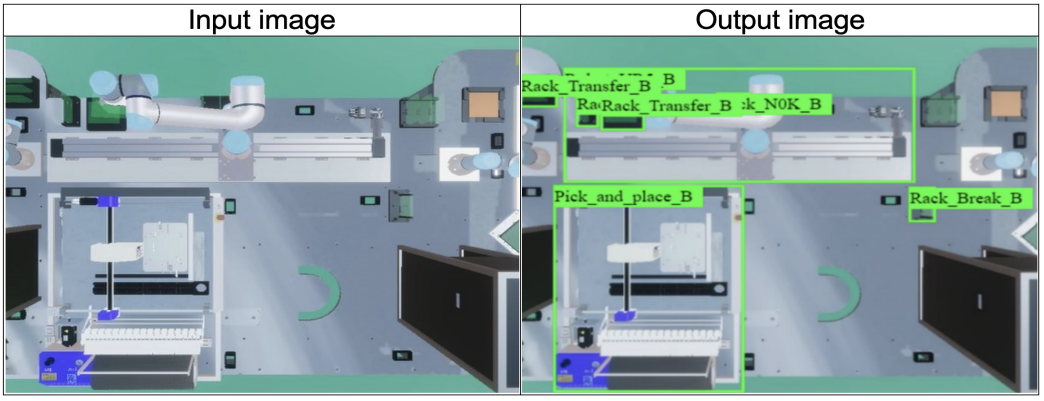
In the fourth and final stage, we obtain the results of the prediction of the image that was made in the previous stage validation of the neural network to verify that the elements of the image have been identified by viewing each element in the image using a box that encloses the element and adds a label to it with the name of the detected element. These validation tests will be necessary to be able to refine the neural network in case tuning is necessary, and that is why one way to validate the correct functioning and behavior of the neural network is to test it using simulated stages to review different cases where the neural network may not be able to detect the elements under certain conditions, or where it can, but the result in the prediction of the image of the elements of the manufacturing cell is not the best, and then the created model of the neural network must be corrected.

3. Results

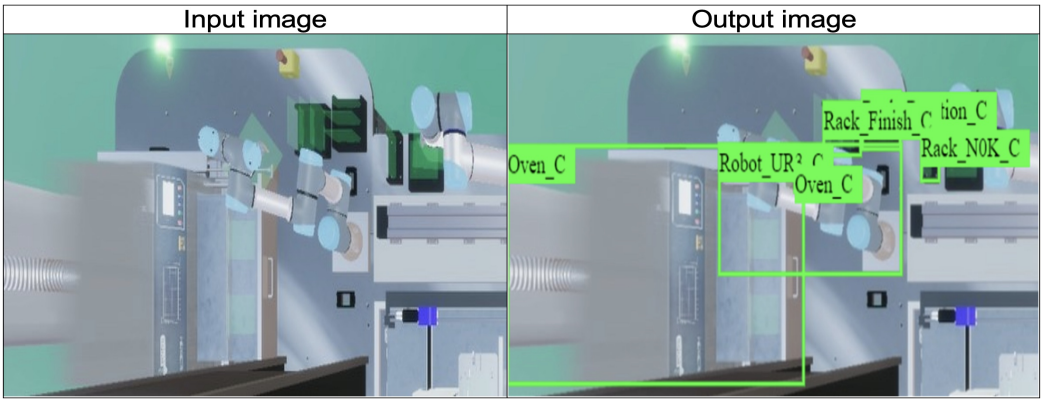
The developed application was tested to identify the elements of the manufacturing cell, and for this, three tests were carried out with different stages, which we will call stages "A, B, and C" (see Figure 5), to identify each case in which the results obtained from the application algorithm are displayed (see Tables 3–5).



(**a**) Validation under stage "A".



(**b**) Validation under stage "B".



(**c**) Validation under stage "C".

Figure 5. Testing the stages "A, B, and C". For each test, an image of the "A, B, and C" area of the manufacturing cell was inserted. Each image turned out different due to the dimension and content displayed of the image selected for each stage.

For the first test, it can be seen that the elements of the stage are enclosed in a box with a label corresponding to the name of the detected element, and the dimension of the inserted image was the original, and in it, all the elements that were expected to be detected in the proposed stage "A" were detected (see Figure 5a). The second test was performed

with an image of stage "B"; the program recognizes the elements and encloses it in a green outline box with its name as the label. Thus, you can see the test image used for stage "B" with the detection of the visible elements (see Figure 5b). In the same way, the third test shows acceptable results in the identification of elements for stage "C" (see Figure 5c).

However, the developed application manages to correctly identify the element, where it is located with precision and accuracy, displaying it in green outline boxes along with the name of the detected element. Before deploying the application to a production environment, tests were carried out in different stages and conditions of manufacturing zone "A" to simulate the possible results when the application is launched in a production environment. Therefore, it will be determined if the elements to be located were as expected, the confidence percentage of the element, and the coordinates at which the element is located in the manufacturing cell space.

The corresponding tests for each stage were carried out with the Postman application, which is a tool that serves as a great help for testing applications in development. Testing the application consisted of inserting a test image and stage type into the Postman request to test the Application Programming Interfaces, and in such a way, inspect the results obtained from image processing with the YOLOv8 model that was trained.

When submitting the POST (to add the information of the detected elements of the image) type request, a JSON (JavaScript Object Notation) [34] is generated from the Postman [35] terminal with the information found during the prediction of the image of the PCB manufacturing cell. Consequently, the values that the terminal will display will be the confidence or probability that the detected element was as expected, the name of the tagged item, and the coordinates at which the detected item was found.

In Figure 6, you can see the test of stage "A" validated with the Postman tool of the elements detected in the image.

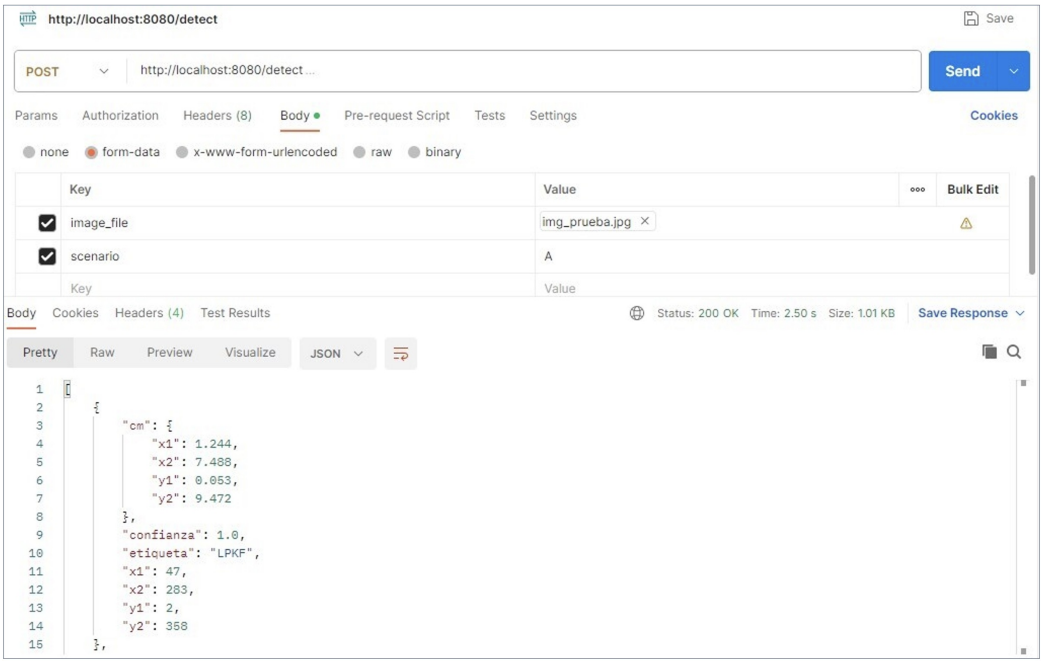


Figure 6. A stage POST request.

Also, we can observe the elements that were detected that are named after the attribute of the tag. However, we have a confidence probability of no less than 94% that the elements detected are as expected. In addition, we have the coordinates of the element detected in "x" and "y" of the prediction of the image of stage "A". Below are the values of the attributes "x1", "y1", "x2", "y2", "Label", and "Trust", where the coordinates in "x1" and "y1" show the coordinates of the detected element and "x2" and "y2" show the dimension

of the detected object, which make up the JSON, referring to each element detected in stage "A" (see Table 3).

Table 3. Results of stage "A".

x1	y1	x2	y2	Label	Trust
47	2	283	358	LPKF	1.0
138	369	176	435	Rack_Inicio	0.99
290	344	380	416	E_Limpieza	0.99
248	533	285	603	Rack_B	0.98
148	497	182	544	Rack_N0K	0.97
314	441	431	608	E_Inspeccion	0.97
87	391	238	513	Robot	0.94

Note: column x1 and y1 form the initial coordinate and column x2 and y2 form the final coordinate of the identified object (label column).

In stage B, as inferred from Table 4, calculating the average confidence of the detected elements shows a mean of 0.97, similar to stage A.

Table 4. Results of stage "B".

x1	y1	x2	y2	Label	Trust
33	178	217	418	Pick_and_place_B	1.0
43	41	384	173	Robot_UR5_B	0.99
192	70	223	87	Rack_N0K_B	0.97
55	70	75	108	Rack_Transfer_B	0.96
380	180	404	219	Rack_Break_B	0.94

Note: column x1 and y1 form the initial coordinate and column x2 and y2 form the final coordinate of the identified object (label column).

In stage C, the average confidence of the detected elements was 0.98. This indicates that the results in Table 5 achieved a slightly higher confidence percentage than the two previous stages.

Table 5. Results of stage "C".

x1	y1	x2	y2	Label	Trust
296	143	330	236	Rack_Finish_C	0.99
332	130	368	225	Rack_Rotation_C	0.99
1	222	278	624	Oven_C	0.99
386	189	403	279	Rack_N0K_C	0.98
198	181	367	427	Robot_UR3_C	0.98

Note: column x1 and y1 form the initial coordinate and column x2 and y2 form the final coordinate of the identified object (label column).

However, it is very important to highlight that the average continues to exceed the 90% threshold, which is positive, as it represents a strong indicator. Therefore, the neural network model used in stage C, unlike stages A and B, has been trained on different images but under the same configuration. This configuration has led to notable accuracy when making inferences and predictions on test images. The high degree of confidence in detecting elements in the manufacturing cell ensures highly reliable and precise results. The application's performance heavily relies on these accurate results because a high confidence percentage assures the validity of the outcomes and the effective operation of the computer vision system over the manufacturing cell elements.

Therefore, we can infer that the development of the application of artificial vision through neural networks for the detection and classification of elements in a manufacturing cell of electronic boards of printed circuits for the tests of the stages with the training of the YOLOv8 model of Ultralytics was satisfactory, since the main objective of the test stages was to obtain a very good inference of the image processing, by obtaining a reduced margin of error from the detected elements of the manufacturing cell.

To achieve this objective, we implemented the YOLOv8 model in the development of the artificial vision system and carefully adjusted the necessary hyperparameters to

establish and deploy the training of the neural network effectively. This allowed us to create a neuron capable of making fast and accurate predictions in images using the new computer vision model, which was trained from a general model specifically adapted to detect elements in the manufacturing environment. In addition, we conduct validation tests in various simulated stages to ensure the efficiency and accuracy of image processing in manufacturing cells.

A neural network-based element detection system optimizes the printed circuit board (PCB) fabrication process in a manufacturing cell (see Figure 7). Specialized models trained for each zone demonstrated the following: the models correctly identified the elements present in each zone, with high confidence percentages; the exact coordinates of each detected element were obtained, allowing its precise location in the manufacturing cell; the dimensions of the detected elements were accurately measured, ensuring an accurate representation of the components.

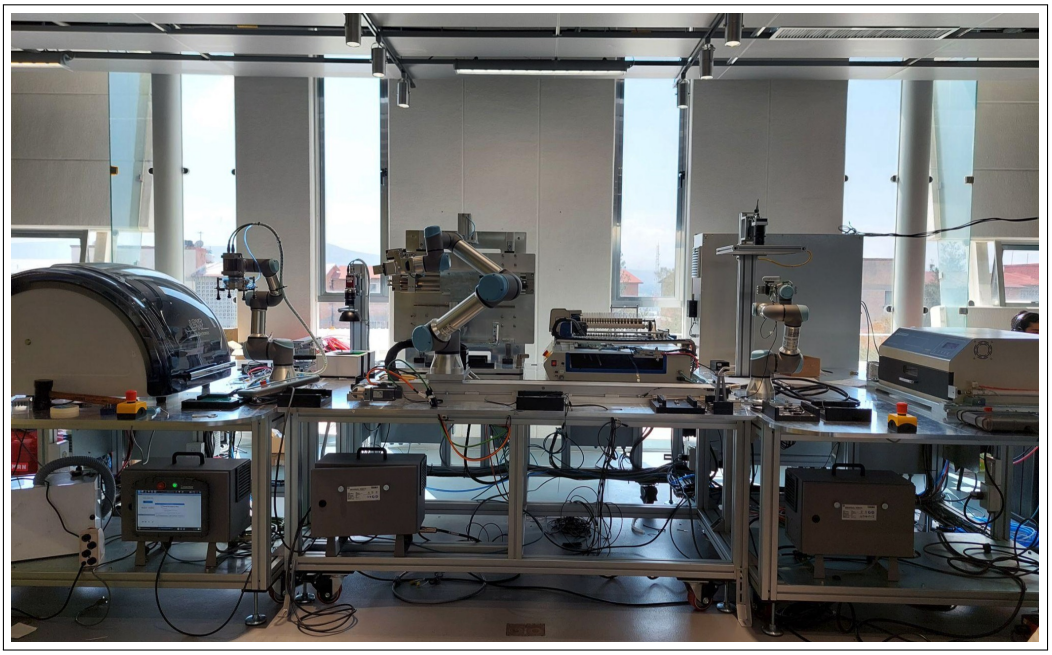


Figure 7. A real printed circuit board manufacturing cell.

4. Discussion

We aim for our case study to be implemented in any type of manufacturing cell that requires flexibility or adaptation due to modifications among the elements of a manufacturing cell, whether in position, orientation, presence, absence, or the addition of unrecognized elements. The system will allow for the early identification of these changes and automatically adjusts the manufacturing process, avoiding failures, unnecessary movements, or actions caused by the mentioned modifications. This approach results in significant energy and time savings by omitting preprogrammed actions that are not appropriate and preventing potential damage to the robots, grippers, PCBs, and other components, which could collide with other elements due to the initially considered changes in the work environment.

5. Conclusions

Testing in the simulated stages gives us a robust assessment of the feasibility and affordability of the model trained for image processing. By providing us with the probability of confidence of the developed algorithm, we can safely infer that the model is suitable for use, since, when calculating the average of the data in stage "A", an average confidence of 0.97 is obtained in the effectiveness of the element detection system. This high degree of confidence in the detection of elements in the manufacturing cell ensures that the results

are highly reliable and accurate in the effective operation of the computer vision system relative to the elements of the manufacturing cell.

In stage B, as inferred from Table 3, calculating the average confidence of the detected elements shows a mean of 0.97, similar to stage A. In stage C, the average confidence of the detected elements was 0.98. This indicates that the results in Table 4 achieved a slightly higher confidence percentage than the two previous stages. Overall, combining the confidence percentages from stages A, B, and C, the model achieved an average confidence of approximately 0.973, reinforcing the reliability and accuracy of the element detection system across different stages.

The ultimate goal of these tests in a simulated environment is to optimize the actual manufacturing cell, reduce cycle times, minimize waste, and ensure product quality. This not only increases productivity but can also have a positive impact on production costs and time savings in manufacturing each printed circuit board. Having previously performed the tests in the simulation environment, we are better prepared to effectively manage and optimize the process in the future.

In manufacturing cells or industrial environments where there is interaction between people and industrial elements, variations can occur, such as changes in element positions, the absence of certain components, or the presence of foreign objects. A system that constantly monitors these variations and provides feedback to robotic arms to update or avoid certain movements or actions can improve process times and detect potential errors in advance. By first performing these actions in simulation to validate their safety, we can ensure that the robotic arms behave appropriately in real life. Future research should incorporate this type of detection in a real environment to validate changes in action execution based on the detected element variations.

Author Contributions: Conceptualization, A.T.-M. and L.B.-R.; methodology, A.T.-M. and M.B.-O.; software, A.T.-M. and M.B.-O.; validation, A.T.-M., M.B.-O. and L.B.-R.; formal analysis, E.A.F.-U. and C.E.C.-G.; investigation, A.T.-M. and M.B.-O.; resources, E.A.F.-U. and C.E.C.-G.; data curation, M.B.-O.; writing—original draft preparation, A.T.-M. and M.B.-O.; writing—review and editing, E.A.F.-U. and A.T.-M.; visualization, A.T.-M.; supervision, E.A.F.-U.; project administration, A.T.-M. and E.A.F.-U.; funding acquisition, E.A.F.-U. and C.E.C.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available at: https://github.com/trejoan1/elementdetection.git (accessed on 26 June 2024). The source code was typed in Python 3 using Visual Studio Code.

Acknowledgments: The authors are grateful for the support received from the CIDESI Research Center, México, and also to the Mexican Science Council CONAHCYT, as the work described in this document is the result of the research project number QID029/QID036.

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

References

1. World Economic Forum. The Fourth Industrial Revolution: What It Means, How to Respond Means, How to Respond. 2016. Available online: https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/ (accessed on 15 June 2024).
2. Perasso, V. BBC Mundo. Qué es la Cuarta Revolución Industrial (y por qué Debería Preocuparnos). 2016. Available online: https://www.bbc.com/mundo/noticias-37631834 (accessed on 13 June 2024).
3. Universidad Autónoma de Madrid. Qué Entendemos por Industria 4.0 o Cuarta Revolución Industrial. 2021. Available online: https://www.uam.es/uam/vida-uam/bibliotecas/biblioteca-politecnica/noticias/la-cuarta-revolucion-industrial (accessed on 13 June 2024).

4. Global PCB Master. Global PCB Manufacturing Industry: Estimated Growth in 2024. 2024. Available online: https://www.pcbmaster.eu/global-pcb-manufacturing-industry-estimated-growth-2024 (accessed on 14 June 2024).
5. JY. Dalian Electronic Co., LTD. Professional PCB & PCBA Manufacturer from China. Whats New? 2019. Available online: https://www.jycircuitboard.com/news/where-are-circuit-boards-made-361.html#:~:text=China%20has%20become%20the%20leader,materials%2C%20and%20advanced%20manufacturing%20technology (accessed on 14 June 2024).
6. Lugaresi, G.; Alba, V.V.; Matta, A. Lab-scale models of manufacturing systems for testing real-time simulation and production control technologies. *J. Manuf. Syst.* **2021**, *58*, 93–108. https://doi.org/10.1016/j.jmsy.2020.09.003.
7. Lugaresi, G.; Matta, A. A remote laboratory experience in teaching discrete event simulation modeling for manufacturing applications. In Proceedings of the Proceedings of the Conference on Learning Factories (CLF), Graz, Austria, 1–2 June 2021. https://dx.doi.org/10.2139/ssrn.3857933.
8. Lugaresi, G.; Loffredo, A.; Roy, S.; Robcis, N.; De Carvalho, V.; Niemeyer, J.F.; Thiede, B.; Di Mascolo, M.; Matta, A. FactoryBricks: a New Learning Platform for Smart Manufacturing Systems. In Proceedings of the Proceedings of the 12th Conference on Learning Factories (CLF 2022), Singapore, 4 April 2022. https://dx.doi.org/10.2139/ssrn.4073311.
9. IBM. What Is Artificial Intelligence (AI)? 2024. Available online: https://www.ibm.com/topics/artificial-intelligence (accessed on 13 June 2024).
10. Oracle. What Is AI? Learn about Artificial Intelligence. 2024. Available online: https://www.oracle.com/artificial-intelligence/what-is-ai/ (accessed on 13 June 2024).
11. Yasnyy-Boyko, I. Aplicación de Algoritmos de Procesado de imáGenes en Dispositivos Reconfigurables. 2023. Available online: https://ebuah.uah.es/dspace/handle/10017/57878 (accessed on 14 June 2024).
12. Reis, D.; Kupec, J.; Hong, J.; Daoudi, A. Real-time flying object detection with YOLOv8. *arXiv* **2023**, arXiv:2305.09972.
13. Bazame, H.C.; Molin, J.P.; Althoff, D.; Martello, M. Detection, classification, and mapping of coffee fruits during harvest with computer vision. *Comput. Electron. Agric.* **2021**, *183*, 106066. https://doi.org/10.1016/j.compag.2021.106066.
14. Chang, S.-J.; Huang, C.-Y. Deep Learning Model for the Inspection of Coffee Bean Defects. *Appl. Sci.* **2021**, *11*; 8226. https://doi.org/10.3390/app11178226.
15. Liu, Z.; Ye, K. YOLO-IMF: An improved YOLOv8 algorithm for surface defect detection in industrial manufacturing field. *Int. Conf. Metaverse* **2023**, *14210*, 15–28. https://doi.org/10.1007/978-3-031-44754-9_2.
16. Amaya-Zapata, S.; Pulgarín-Velásquez, D.; Torres-Pardo, I.D. Development and Implementation of an Artificial Vision System Based on Free Use Languages for the Coach System Products on the Integrated Manufacturing Center (IMC). *Lámpsakos* **2016**, *15*, 43–50. http://dx.doi.org/10.21501/21454086.1702.
17. Abdullah-Al-Noman, M.; Eva, A.N.; Yeahyea, T.B.; Khan, R. Computer vision-based robotic arm for object color, shape, and size detection. *J. Robot. Control (JRC)* **2022**, *3*, 180–186. https://doi.org/10.18196/jrc.v3i2.13906.
18. Du, Y.C.; Muslikhin, M.; Hsieh, T.H.; Wang, M.S. Stereo vision-based object recognition and manipulation by regions with convolutional neural network. *Electronics* **2020**, *9*, 210. https://doi.org/10.3390/electronics9020210.
19. Renteria-Vidales, O.I.; Cuevas-Tello, J.C.; Reyes-Figueroa, A.; Rivera, M. ModuleNet: a convolutional neural network for stereo vision. In Proceedings of the Pattern Recognition: 12th Mexican Conference, MCPR 2020, Morelia, Mexico, 24–27 June 2020; Proceedings 12; Springer: Berlin/Heidelberg, Germany, 2020; pp. 219–228. https://doi.org/10.1007/978-3-030-49076-8_21.
20. Son, Y.H.; Park, K.T.; Lee, D.; Jeon, S.W.; Do, N.S. Digital twin–based cyber-physical system for automotive body production lines. *Int. J. Adv. Manuf. Technol.* **2021**, *115*, 291–310. https://doi.org/10.1007/s00170-021-07183-3.
21. Icarte-Ahumada, G.A. Aplicaciones de inteligencia artificial en procesos de cadenas de suministros: Una revisión sistemática. *Ingeniare. Rev. Chil. Ing.* **2016**, *24*, 663–679. https://doi.org/10.4067/S0718-33052016000400011.
22. Dwyer, B.; Gallagher, J. Getting Started with Roboflow. 2023. Available online: https://blog.roboflow.com/getting-started-with-roboflow/ (accessed on 10 June 2024).
23. Londoño, P. Qué es Python, Para qué Sirve y Cómo se Usa (+ Recursos Para Aprender). 2023. Available online: https://blog.hubspot.es/website/que-es-python (accessed on 10 June 2024).
24. VanderPlas, J. TensorFlow in Google Colaboratory. 2021. Available online: https://colab.research.google.com/ (accessed on 10 June 2024).
25. Glenn, J.; Ayush, C. Ultralytics. 2023. Available online: https://ultralytics.com/ (accessed on 11 June 2024).
26. Glenn, J.; Ayush, C. Introducing Ultralytics YOLOv8. 2023. Available online: https://docs.ultralytics.com/es (accessed on 11 June 2024).
27. OpenCV. Introduction to OpenCV. 2024. Available online: https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html (accessed on 19 June 2024).
28. Lambda. The GPU Cloud for AI – On–Demand NVIDIA GPU Instances & Clusters for AI Training & Inference. San Jose, CA, EE.UU. 2024. Available online: https://docs.opencv.org/4.x/d6/d00/tutorial_py_root.html (accessed on 28 June 2024).
29. Pichai, S. Google Chrome – The browser built to be. Mountain View, CA, EE.UU. 2024. Available online: https://www.google.com/chrome/ (accessed on 28 June 2024).
30. Microsoft Visual Studio Code - Code Editing. Redefined. Redmond, Washington, EE.UU. 2024. Available online: https://www.google.com/chrome/ (accessed on 28 June 2024).
31. Glenn, J.; Burhan, Q.; Laughing, Q. Instance Segmentation, YOLOv8n-seg. 2024. Available online: https://lambdalabs.com (accessed on 28 June 2024).

32. Selcuk, B.; Serif, T. A Comparison of YOLOv5 and YOLOv8 in the Context of Mobile UI Detection. In Proceedings of the International Conference on Mobile Web and Intelligent Information Systems, Marrakech, Morocco, 14–16 August 2023; pp. 161–174. https://doi.org/10.1007/978-3-031-39764-6_11.
33. Fernández, Y. API: Qué es y Para Qué Sirve. 2019. Available online: https://www.xataka.com/basics/api-que-sirve (accessed on 10 June 2024).
34. NextU, L. What is Json? Why Is It Important to Know it? 2022. Available online: https://www.nextu.com/blog/que-es-json-por-que-es-importante-conocerlo-rc22/ (accessed on 19 June 2024).
35. Muradas, Y. Qué es Postman y Primeros Pasos. 2019. Available online: https://openwebinars.net/blog/que-es-postman/ (accessed on 19 June 2024).