Article

SolDef_AI: An Open Source PCB Dataset for Mask R-CNN Defect Detection in Soldering Processes of Electronic Components

Gianmauro Fontana [1] , **Maurizio Calabrese** [2,*] , **Leonardo Agnusdei** [2] , **Gabriele Papadia** [2]
and Antonio Del Prete [2]

1   Institute of Intelligent Industrial Technologies and Systems for Advanced Manufacturing CNR-STIIMA, Via A. Corti 12, 20133 Milan, Italy; gianmauro.fontana@stiima.cnr.it

2   Department of Innovation Engineering, University of Salento, Via per Monteroni, 73100 Lecce, Italy; leonardo.agnusdei@unisalento.it (L.A.); gabriele.papadia@unisalento.it (G.P.); antonio.delprete@unisalento.it (A.D.P.)

*   Correspondence: maurizio.calabrese@unisalento.it

Abstract: The soldering process for aerospace applications follows stringent requirements and standards to ensure the reliability and safety of electronic connections in aerospace systems. For this reason, the quality control phase plays an important role to guarantee requirements compliance. This process often requires manual control since technicians' knowledge is fundamental to obtain effective quality check results. In this context, the authors have developed a new open source dataset (SolDef_AI) to implement an innovative methodology for printed circuit board (PCB) defect detection exploiting the Mask R-CNN algorithm. The presented open source dataset aims to overcome the challenges associated with the availability of datasets for model training in this specific research and electronics industrial field. The dataset is open source and available online.

Keywords: open source dataset; SMT component soldering; aerospace printed circuit board manufacturing; printed circuit board defect detection; automated optical inspection; deep learning; mask R-CNN

1. Introduction

Electronic printed circuit boards (PCBs) play a pivotal role in the modern world of satellite technology, particularly in the field of microsatellites. These devices have revolutionized space exploration by offering a cost-effective and agile platform for various missions, from Earth observation to scientific research. At the heart of every microsatellite, intricate PCBs contain and connect different electronic components responsible for communication, navigation, data collection, etc.

PCBs for microsatellites are characterized by their compact design, resilience to radiation, ability to operate in extreme temperatures, durability against mechanical stress during launch, efficient power consumption, hermetic sealing for space protection, high-frequency communication capabilities, redundancy for mission reliability, long lifespan, and adaptability to mission-specific requirements. These features collectively enable microsatellites to conduct diverse space missions while withstanding the challenging conditions of outer space.

The production of PCBs for microsatellites is a highly specialized and precise endeavor. These PCBs must not only withstand the harsh conditions of outer space, including extreme temperatures, radiation, and vacuum, but also be meticulously designed and assembled to ensure the reliability and longevity of the satellite's mission. In this overview, we delve into the fascinating world of electronic PCBs for microsatellites, exploring their crucial role in space exploration and the intricacies of their production process.

The production of PCBs for microsatellites is subject to strict regulations and standards to ensure the reliability and safety of the electronic components operating in space. Some

of the key reference standards include IPC-A-610, NASA-STD-8739.3, ECSS-Q-ST-70-38C, MIL-PRF-31032, and MIL-PRF-55110.

These regulations provide clear guidelines for the design, production, and quality control of PCBs for microsatellites. By ensuring compliance with these standards, it is possible to ensure that these electronic boards are suitable for the space environment's challenges and critical space missions' needs.

For the specific application of SMT soldering, the reference regulation is ECSS-Q-ST-70-38C.

This standard was developed by the European Space Agency (ESA), and it concentrates on technical specifications for the assembly of space electronic components, including PCBs. It covers aspects such as materials, processes, inspection, and testing.

For these reasons, the detection of defects in the soldering process of micro/meso electronic components on PCBs is crucial for ensuring the quality and reliability of electronic devices. Several research works have been conducted to develop effective defect detection techniques, focusing on leveraging advanced technologies such as deep learning, image processing, and artificial intelligence [1–9]. In [10], deep autoencoders were used for defect detection, emphasizing the specialized nature of soldering defect detection and the application of traditional image processing techniques. Similarly, in [11], a genetic algorithm and a neural network were combined to enhance automatic optical inspection for solder connection defects in surface mount components on PCBs. The model was supported by the AdaMax optimization method, which was highly efficient. The data obtained showed that, compared to the RMSprop and Adam methods, the AdaMax optimization method was more efficient in achieving results with a 96.40% success rate in 100 epoch loops. These studies underscore the significance of advanced technologies in improving defect detection processes.

Furthermore, in [12], the potentiality of a model developed by the authors was demonstrated through extensive experiments on a PCB dataset (i.e., FSOD professional few-shot object detection dataset created by Tencent in 2020), indicating the potential for innovative approaches to defect detection. In [13], the authors also focused on utilizing deep learning, specifically a skip-connected convolutional autoencoder, for PCB defect detection, emphasizing the criticality of surface inspection in ensuring quality control. In addition, in [14,15], the importance of image processing algorithms was highlighted in detecting soldering defects and soldering joints in the PCB defect detection process. These references emphasize the role of image processing in identifying and classifying defects, contributing to the overall quality control process. Moreover, in [16], the authors introduced an improved IoU-based loss function for PCB defect detection, showcasing the potential for enhancing existing algorithms to achieve higher accuracy. In [17], a novel automated defect detection approach was proposed for the PCB manufacturing process, indicating ongoing efforts to innovate defect detection methodologies.

The study in [18] introduces the TDD-net, a tiny defect detection network specifically designed for PCBs. Quantitative results on the PCB defect dataset show that the proposed method has better portability and can achieve 98.90% mean average precision (mAP), which outperforms the state of the art. They applied an open PCB defect dataset to verify their PCB defect detection method, emphasizing the practical application of their approach. Additionally, they proposed an improved IoU-based loss function for PCB defect detection, showcasing the potential for enhancing existing algorithms to achieve higher accuracy.

The study in [19] adopted YOLOv4-MN3 for PCB surface defect detection, focusing on automatically locating small and dense solder joints in PCB images. The experimental results show that the improved detector achieved a high performance, scoring 98.64% mAP at 56.98 frames per second (fps), outperforming the other compared to SOTA detectors. In [20], the D3PointNet model was proposed for dual-level defect detection in solder paste printers, emphasizing defect region proposal and classification. Experimental results showed that the proposed D3PointNet is robust to the sparsity and size changes of the DSPP image, and its exact match score was 10.2% higher than that of the existing CNN-

based state-of-the-art multi-label classification model in the DSPP image dataset. In [21], the authors highlight the significance of semi-supervised defect detection methods with data-expanding strategies to address the challenges of labeling large-scale datasets for PCB quality inspection. The experimental results on DeepPCB indicate that the proposed DE-SSD achieves a state-of-the-art performance, with an improvement in mAP of at least 4.7 compared with previous methods. A novel PCB defect detection method based on digital image processing was introduced in [22], demonstrating the effective detection and analysis of solder pads and solder pastes.

Experiments show that this method can effectively detect and analyze various information about solder pads and solder pastes, among which the detection accuracy rate of solder pads is 99.4% and the missed detection rate is 0.4%. The detection accuracy rate of solder paste is 99.3%. The work in [23] develops the LPViT model for PCB image classification and defect detection, focusing on the DeepPCB dataset containing various PCB defects. LPViT improves all metrics by more than 6%. It also outperforms the current SOTA model (98.8% vs. 98.6%). The study in [24] proposes an improved YOLOv5-based model for automatic PCB defect detection, emphasizing the training of models using artificial defect image datasets. The results revealed that this model performed well on defect location and classification, with its mAP0.5 reaching 63.4%. The work in [25] introduces the PCB-YOLO algorithm, an improved detection method based on YOLOv5 for PCB surface defect detection, addressing issues of accuracy and speed. The experimental results show that PCB-YOLO achieves a satisfactory balance between performance and consumption, reaching 95.97% mAP at 92.5 fps. A preliminary study was proposed in [26] to analyze the ML capabilities to perform automated optical inspection (AOI) for quality control in the manufacturing of PCBs. The target was to investigate the performance of the Mask R-CNN method in identifying the main PCB defects after the manufacturing process.

While there are several other studies on defect detection in PCBs including the use of deep learning models and advanced algorithms, the selected references align closely with the focus on innovative defect detection methodologies and the application of open PCB datasets to validate the proposed approaches.

These references collectively highlight the ongoing advancements in defect detection for PCBs, emphasizing the significance of leveraging advanced technologies and datasets to enhance the reliability and quality of electronic components.

Some open source datasets are available in the literature for benchmarking machine vision-based PCB inspection techniques. Few of these datasets mimic real-world scenarios essential for effective automated optical inspection (AOI) and classification methods. Inspections may focus on the PCB assembly (PCBA), where all the components have been placed and soldered, for categorizing PCBs/components and assessing the soldering and assembling quality, or solely on the "naked" PCB to identify manufacturing defects.

Table 1 presents a comparison of the more relevant available PCB datasets.

In this context, the dataset presented in this manuscript was designed and created since none of the datasets available in the literature are related to the production of PCBs for aerospace applications. Therefore, none of these datasets have been designed for the soldering and assembly defects mentioned in the specific reference standards that regulate PCB assemblies for aerospace systems, such as ECSS-Q-ST-70-38C.

Therefore, the proposed work aims to make available an open source dataset, i.e., SolDef_AI, to give the possibility to carry out other research in defect detection for PCB products considering the difficulty of having a collection of images for this specific sector to train AI models.

Table 1. Comparison of more relevant available PCB datasets.

Datasets	Number of Images	Inspected Items	Inspected Target	Defect Classes	Vision System	Resolution [Mpx]	Spatial Resolution [px/cm]	Image Variation
FICS-PCB [27]	9.912	6 SMT components	Component classification	-	Digital microscope + DSLR camera	10 + 45.7	462 − 921 + 118	Lighting intensity and image scale
DeepPCB [28]	1500	PCB traces	Defect identification	6	CCD camera	25.6	480	no
PCBA-def [29]	1386	PCB traces	Defect identification	6	Digital microscope	16.2	n.a.	PCB rotation
PCB-Metal [30]	984	4 SMT components	Component classification	-	DSLR camera	30.4	n.a.	PCB rotation
PCB-DSLR [31]	748	1 SMT components	Component classification	-	DSLR camera	16.2	87.4	PCB rotation
Proposed	1115	4 SMT components + soldered joints	Defect identification	2 + 4	Digital microscope	5	3333–6536	Lighting intensity and point of view

2. The SolDef_AI Dataset

We contribute the SolDef_AI dataset to the community, containing 1150 soldered SMT component images covering defective and defect-free assemblies. Each soldered SMT component was acquired from three different points of view to provide a global view of the soldered component and its solder joints on the PCB. Two classes of SMT assembly defects were collected: the first includes defects related to the incorrect position of the soldered component on the PCB (e.g., misalignment of SMT component with respect to the PCB assembly pads), and the second is defects related to defective solder joints on the PCB (e.g., excessive or insufficient quantity of soldering material on the joint).

2.1. Dataset Acquisition

To ensure the representativeness of the dataset, a PCB image acquisition system was implemented that resembles the optical inspection system used manually by the operator in the PCB assembly process for the aerospace field.

Therefore, following the standard industrial settings, the vision system was designed to acquire high-quality images of soldered SMT components on PCBs in a repeatable way. The resolution of the system was chosen to acquire the smallest SMT component soldered on the test PCBs and the smallest solder joints on the PCB pads. Additionally, these images are free from effects such as blur and poor lighting and were taken from PCBs clear of dust and other debris. The images in this dataset were obtained from different PCBs used to train operators in soldering SMT components.

Figure 1 shows the vision system used for the dataset acquisition. It includes a portable USB low-cost microscope by Cainda with an HD color CMOS sensor, a resolution of 2560 × 1440 pixels, 1000× maximum magnification, and a frame rate of 30 f/s [32]. A knob can be used to set the magnification on the microscope body. The setup also includes a lighting system consisting of an LED ring with a diffusor that allows a 4-level adjustable light control. The microscope and the lighting unit, mounted on a flexible arm, overlook a manual XY linear stage where the PCBs are placed for the acquisition. The XY linear stage allows translation of the PCB along two orthogonal axes to acquire the different SMT components soldered on its pads. The flexible arm holding structure allows for changing the position of the vision system to acquire the PCB and SMT components from various points of view. The camera and the light were controlled simultaneously using a PC-based software developed in Labview 2021. The lighting system ensures diffuse illumination, thus avoiding shadows and specular reflections on the board caused by the environmental lighting.
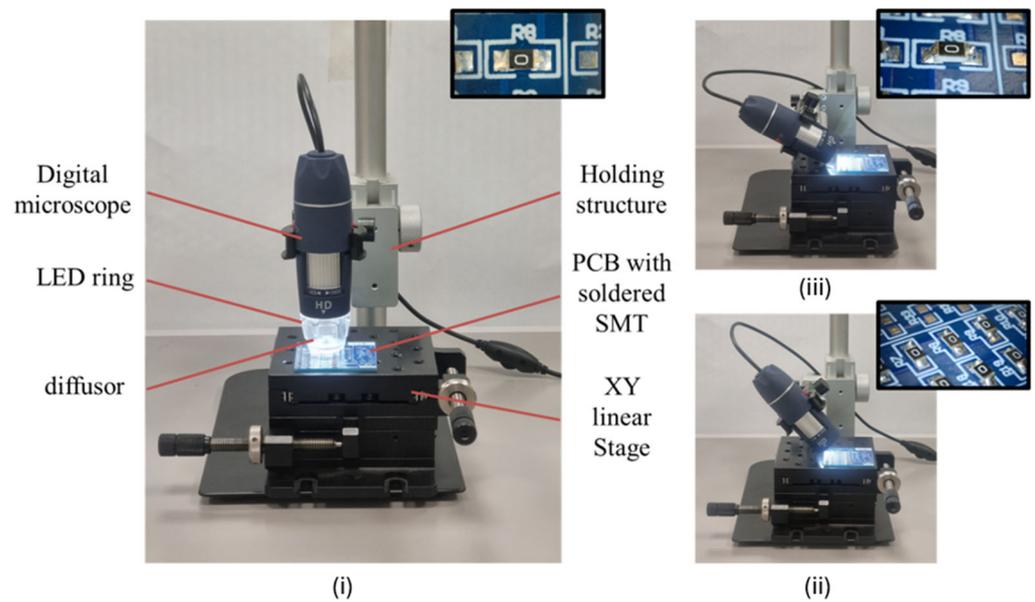
Figure 1. The vision system designed to acquire images for the SolDef_AI dataset. Setup for (**i**) top view, (**ii**) 45-degree view, and (**iii**) axonometric view of the soldered SMT components.

The dataset contains images of various SMT components soldered on different PCBs. Each component was acquired from three points of view: top (see Figure 1i), 45 degrees (see Figure 1ii), and axonometric view (see Figure 1iii). These configurations enable a complete visualization of all details of the soldered component from all its sides to support the identification of its position on the PCB (from the images with the top view) and the status of its solder joints on the PCB (from the images with 45-degree and axonometric views).

Since the dimensions of the various SMT components are different, the magnification value and the working distance of the vision system were adjusted for the acquisitions of each component to increase the spatial resolution and show more details, i.e., the smallest component was set with the highest magnification.

The vision system was calibrated to compute image pixel to real-world unit transformation and to compensate for perspective, distortion, and spatial referencing errors [33]. For the current setup, a grid of black dots on a white plate was used to perform the 2D calibration of the vision system (Low Reflect Grid Distortion Target 3" × 3", 0.25 mm dot by Edmund Optics, Barrington, NJ, USA).

According to the chosen setup, the field of view (FoV) was 3.92 × 2.205 mm for the smallest SMT component (i.e., 0603 resistor in imperial unit) with a spatial resolution (Rs) of 1.53 μm/pixel, which allows for a feature resolution (Rf) of 4.6 μm considering 3 pixels spanning the minimum size feature and a measurement resolution (Rm) of 0.153 μm since the measurement resolution capability in pixels (Mp) is 1/10 (i.e., sub-pixel capability of the used machine vision algorithms) [34]. For the biggest component (i.e., 1206 resistor), the FoV was 7.89 × 4.44 mm with an Rs of 3.0 μm/pixel, Rf of 9 μm, and Rm of 0.3 μm.

Figure 2 shows the 6 PCBs where the soldered SMTs were acquired to create the SolDef_AI dataset.
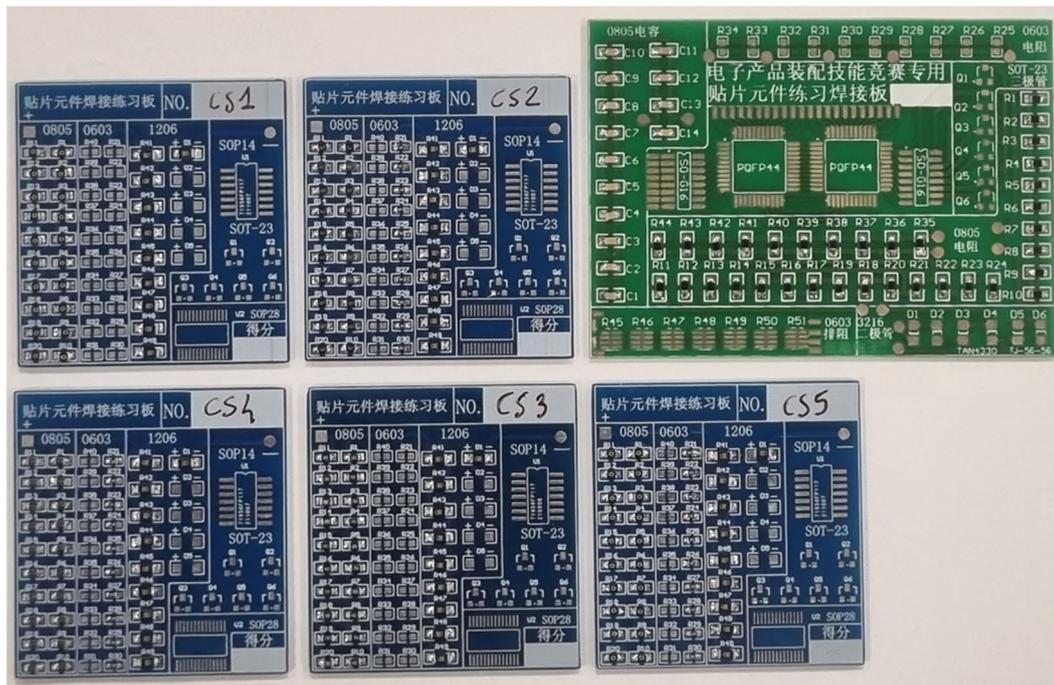
Figure 2. This is a figure. Schemes follow the same formatting.

2.2. Dataset Statistics

The dataset includes images of 15 0805-type capacitors and 30 0603-type, 135 0805-type, and 50 1206-type resistors, including their solder joints with the PCB pads. It can be observed that the dataset is not balanced in the number of images for each electronic component type. This choice ensured that the model was not biased towards a specific type of microelectronic component but generalized to any component with a similar shape. Since the different components in the datasheet have two leads, the dataset includes images of 460 solder joints. Considering that each component was acquired from three different points of view and with different lighting conditions and directions, the total number of images in the dataset is 1150.

Two types of SMT assembly defect categories are presented in the dataset images, as well as SMT components soldered correctly: the first includes defects related to the incorrect position of the soldered SMT on the PCB, i.e., misalignment of SMT component with respect to the PCB assembly pads; the second is defects related to the incorrectly soldered joint of the SMT, i.e., excessive or insufficient quantity of soldering material and presence of spikes on the joint.

In addition to these raw images, for a subset of 228 top view acquired images, the dataset provides manually annotated bounding polygon locations (xy points in pixel) of the image area that includes the component and the assembly pads on the PCB, with an associated class binary ID to identify whether the SMT position on the PCB is correct or not. Moreover, for a subset of 200 45-degree view acquired images, the bounding polygon locations (xy points in pixel) of the image areas that include solder joints were annotated with a class ID for the status of each of them (i.e., correct, excessive, insufficient quantity of solder material, and presence of spikes), as explained in Section 3. The annotations were quality checked by an external reviewer, i.e., a trained person who did not perform the original annotation.

The images and the annotations can be used to train and test object detection techniques that would identify and localize these components with respect to the assembly pads on the PCB to check the correctness of the SMT assembly positions on the board. Moreover, this dataset can be used to train and test object detection techniques that check the correctness of the solder joints on the boards. These annotations are available in a JSON

file with a standard text format that most object/defect detection and classification machine learning algorithms can readily manage.

Tables 2–5 present the statistics of the designed dataset.

Table 2. The distribution of position defects per SMT component.

SMT Types	SMT Number	Correct Position Number	Wrong Position Number
C 0805	15	10	5
R 0603	30	20	10
R 0805	135	90	45
R 1206	50	34	16
Total	230	154	76

Table 3. The distribution of SMT components in top view images.

SMT Types	Top View Image Number	
	Setup 1 (Brighter Lighting Condition)	Setup 2 (Darker Lighting Condition)
C 0805	15	15
R 0603	30	30
R 0805	135	135
R 1206	50	50
Total	460	

Table 4. The distribution of soldering defects per SMT component type.

SMT Types	Soldered Joint Number	Correct Soldered Joint Number	Joints with Excessive Quantity of Solder Material	Joints with Insufficient Quantity of Solder Material	Joints with Presence of Spike
C 0805	30	15	5	5	5
R 0603	60	30	10	10	10
R 0805	270	190	60	60	60
R 1206	100	40	20	20	20
Total	460	275	95	95	95

Table 5. The distribution of SMT components in 45-degree and axonometric view images.

SMT Types	45-Degree View Image Number		Axonometric View Image Number
	Setup 1 (Top–Bottom Direction)	Setup 2 (Bottom–Top Direction)	
C 0805	15	15	15
R 0603	30	30	30
R 0805	135	135	135
R 1206	50	50	50
Total	460		230

3. Labeling Strategies

This paper aims to monitor the soldering process of the microelectronic components on printed circuit boards (PCBs). The most common defects of these processes are related to the position of the components [35,36] (Figure 3) and the quantity of the soldering material [37] (Figure 4). In Figure 4a, a perfect solder joint is represented, and Figure 4b highlights an example where the amount of solder material is excessive (convex joint), while Figure 4c represents a case where the amount of solder material is insufficient (concave joint). Figure 4d presents an example of a solder joint with spikes, showing the presence of protrusions caused by solder material disengagement.

1) Electrical contacts of components; 2) Pads for solder; 3) Component

(**a**)   (**b**)
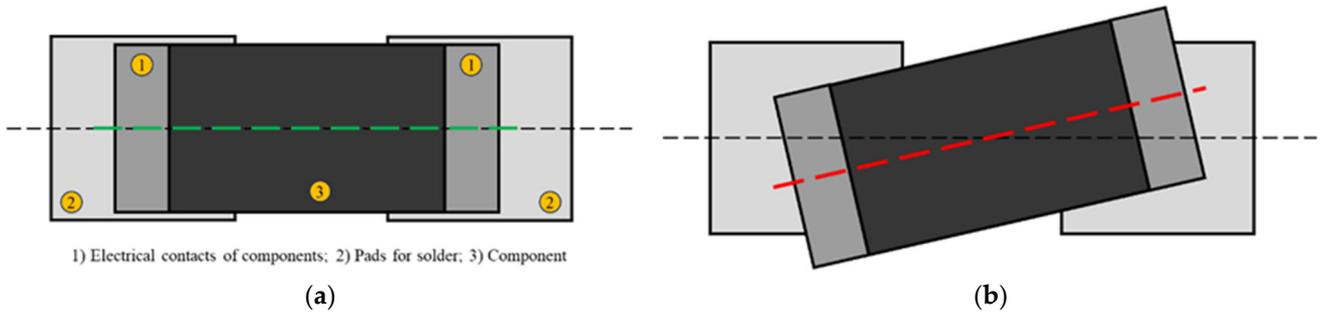
Figure 3. Top view of the component: (**a**) correctly soldered on the PCB; (**b**) not correctly soldered on the PCB.

(**a**)   (**b**)   (**c**)   (**d**)
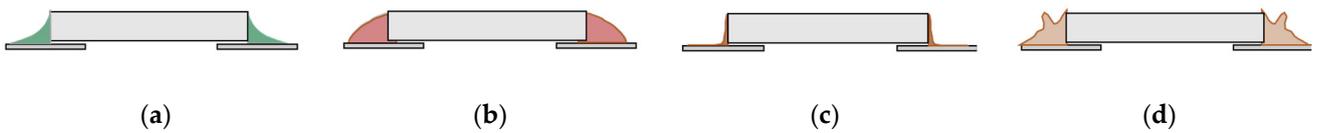
Figure 4. Lateral view of components with (**a**) the correct quantity of solder material, (**b**) an excessive quantity of soldering material, (**c**) an insufficient quantity of soldering material, and (**d**) the presence of spikes.

In order to ensure the correct prediction of the instances, two different datasets were built, one for each task. In particular, dataset_1 includes images from the top view related to the SMT positioning, while dataset_2 contains images from the 45-degree view related to the quantity of the soldering material of joints. Figures 5 and 6 show two representative images of dataset_1 and dataset_2.
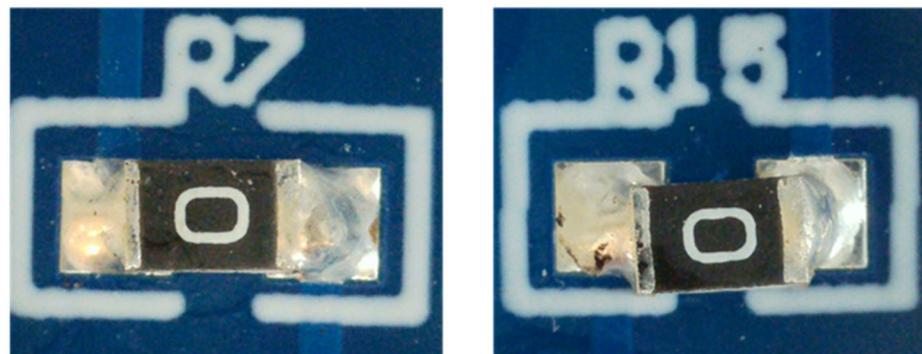
Figure 5. Sample of images included in dataset_1.
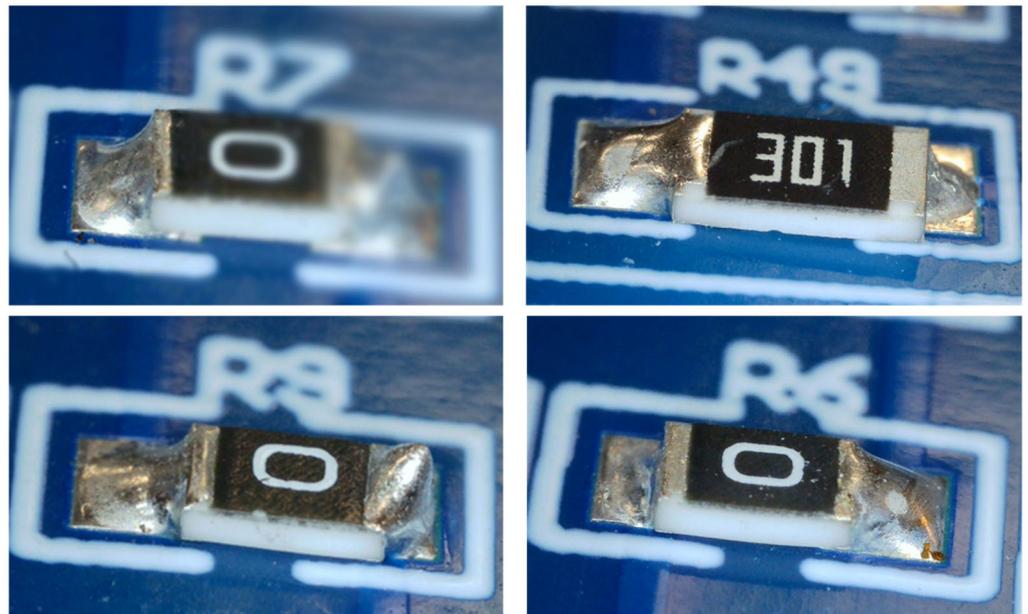
Figure 6. Sample of images included in dataset_2.

Different instances were manually annotated using an open source image annotation tool called LabelMe. This tool allowed for the precise delineation of the footprint for each instance and assigning them specific labels. By utilizing LabelMe, it was possible to generate a JSON file containing all the created masks for each instance.

For dataset_1, the labeling assumption was made by creating rectangular masks, including the two pads and the entire component in the case of good placement, while for incorrect positioning, a polygonal shaped mask was generated, including the same elements. Figures 7 and 8 provide a representation of the labeling approach adopted for the wrong positioning of the component on the PCB.
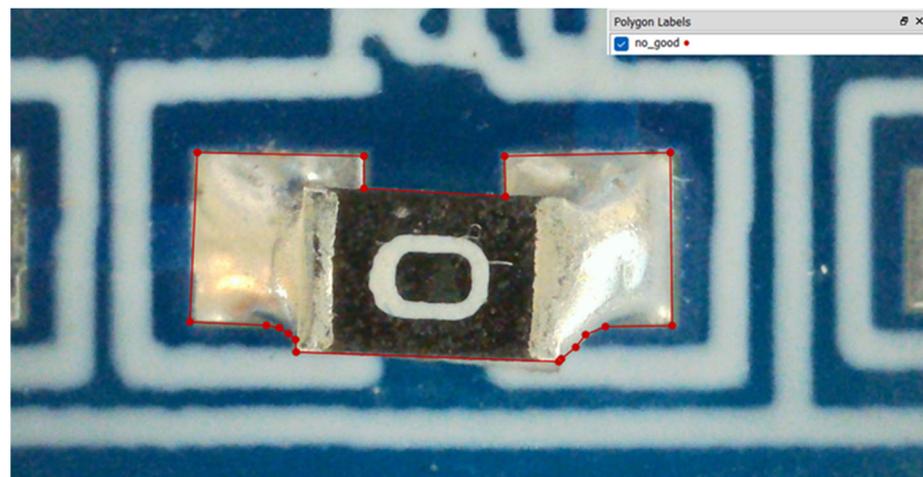
Figure 7. Incorrect positioning of the component (1)—no_good label.
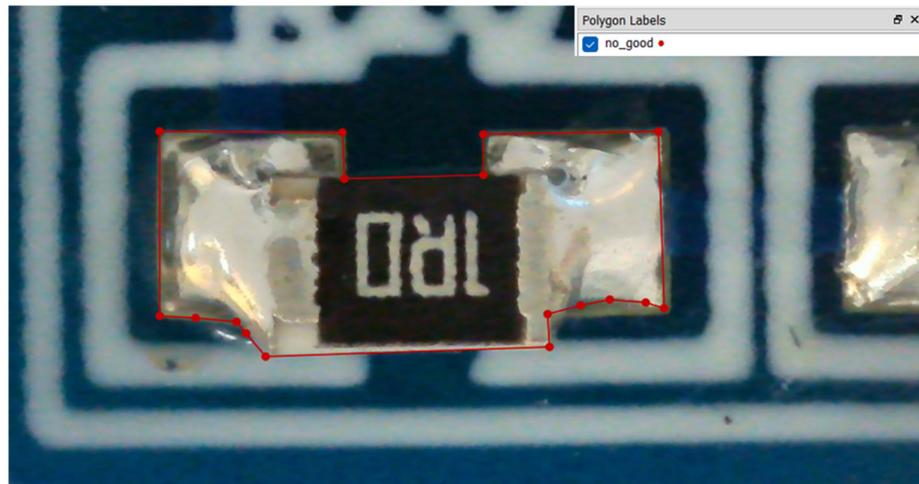
Figure 8. Incorrect positioning of the component (2)—no_good label.

Figures 9 and 10 provide a representation of the labeling approach adopted for the correct positioning of the component on the PCB.

Figure 9. Correct positioning of the component (1)—good label.
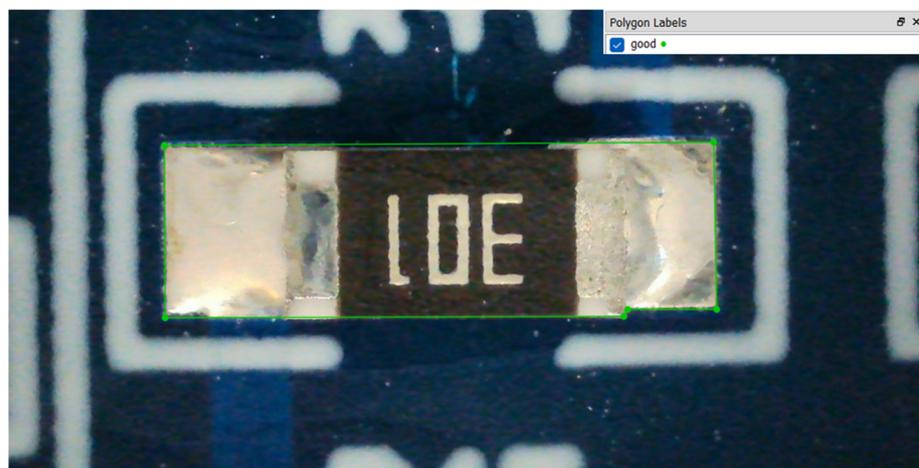
Figure 10. Correct positioning of the component (2)—good label.

A different methodology was employed to monitor joints regarding solder material quantity and the presence of spikes. The labeling approach involved the creation of masks with distinct profiles for each instance. Specifically:

- For properly executed solder joints, a mask with an upward concavity and a specific area was built (Figure 11);
- For solder joints characterized by an excessive amount of material, a mask with a downward concavity was built (Figure 12);
- For joints with insufficient solder material, a mask with a downward concavity was built but with a lower area in comparison to the properly executed solder joint (Figure 13);
- For solder joints characterized by spikes, a mask with a protrusion on its profile was generated (Figure 14).
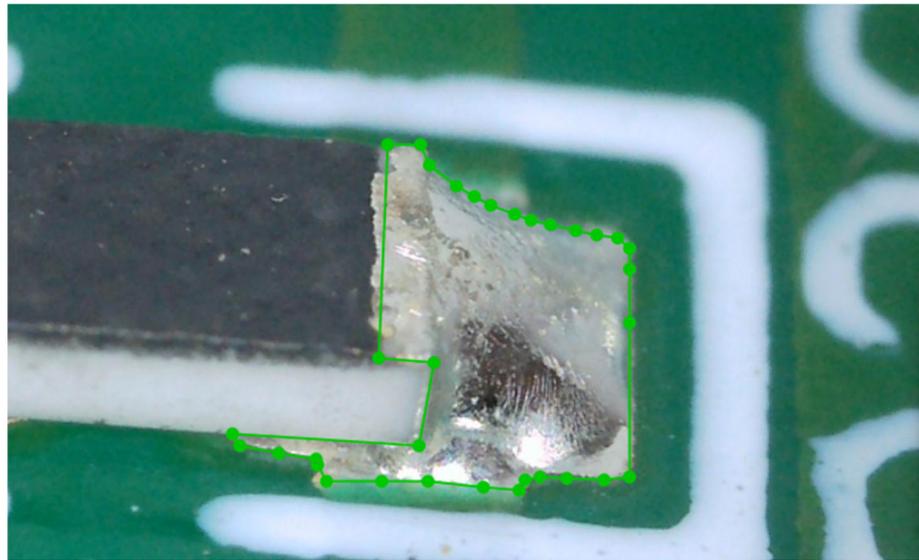
Figure 11. Labeling of a good solder joint.
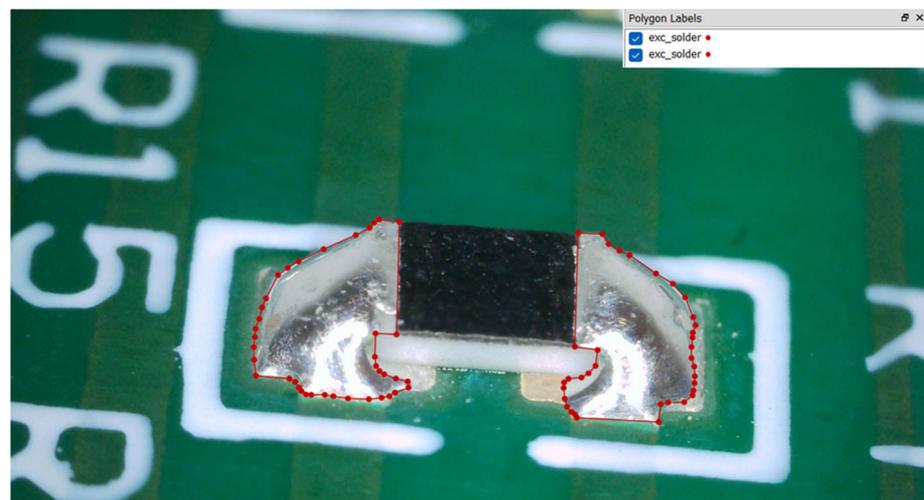
Figure 12. Labeling of an excessive solder joint.
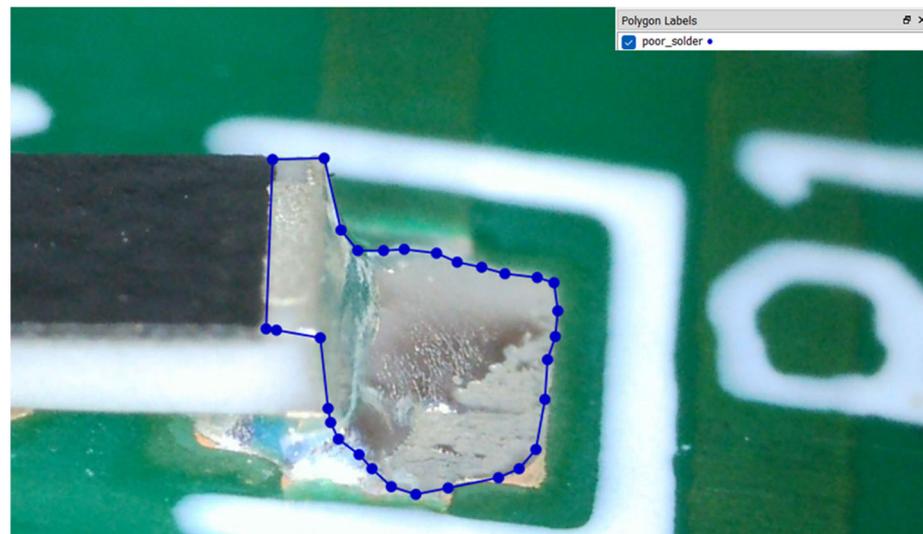
Figure 13. Labeling of a poor solder.
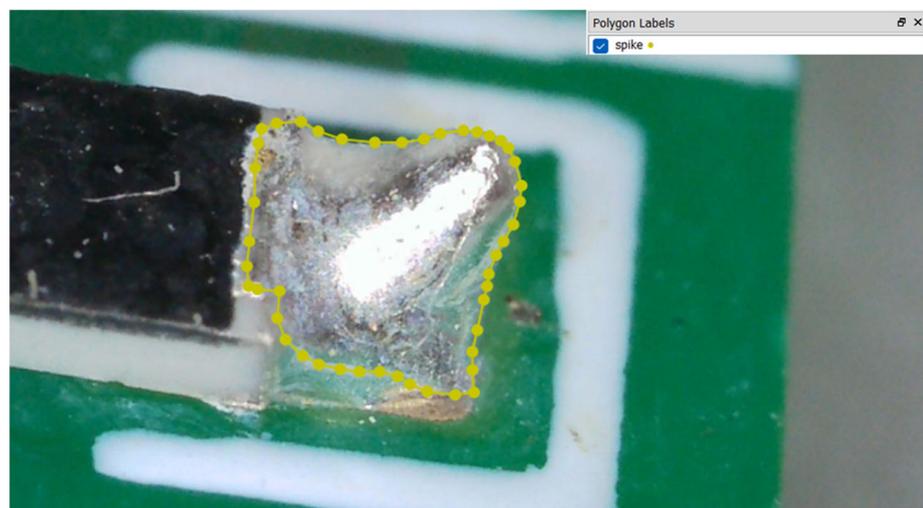
Figure 14. Labeling of a solder joint with spike.

4. Model Configuration

This study used a convolutional neural network called Mask-RCNN (short for mask region-based convolutional neural network). This network extends the Faster RCNN by incorporating an additional branch for predicting pixel-level segmentation masks for each detected object. It is suitable for the application studied in this paper since it integrates both object detection and instance segmentation tasks. The architecture consists of three main components: a convolutional backbone for feature extraction, a region proposal network (RPN) for suggesting regions of interest (ROIs) for object segmentation, and a box head and ROI head for selecting proposed regions and refining their boundaries for more precise segmentation. The implementation utilized the Detectron2 library. A visual representation of the Mask-RCNN architecture is depicted in Figure 15.

In Mask R-CNN configuration, hyperparameter settings are crucial aspects influencing the behavior of the Mask R-CNN model during both the training and inference stages. These hyperparameters, including learning rate, batch size, and number of iterations, significantly influence the model's performance. In Table 6, the specific hyperparameters are reported and detailed.
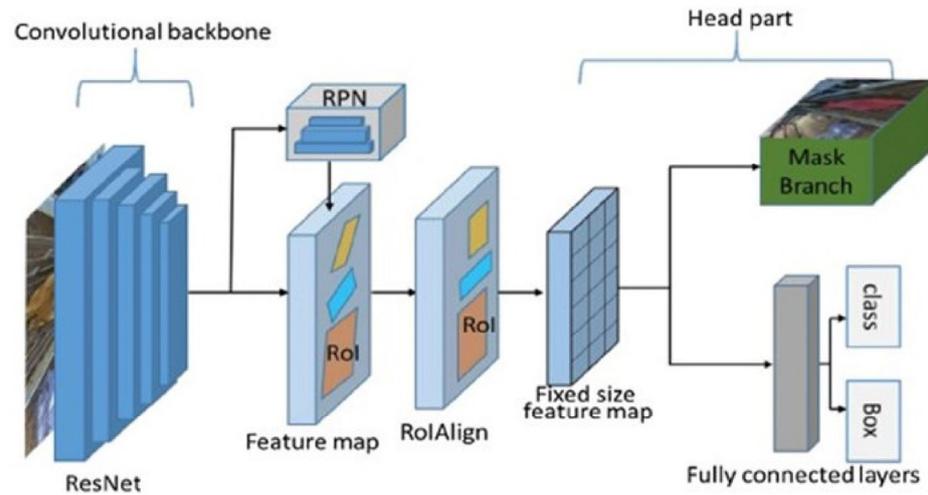
Figure 15. Mask R-CNN configuration.

Table 6. Model hyperparameters.

Variables	Setting
Batch size	128
Learning rate	0.001
Learning rate schedule	Adagrad
Rotation	None
Weight decay	None

5. Model Evaluation

Evaluation of the CNN performance was carried out using the average precision (AP) metric. Furthermore, the model's ability to identify defect locations was assessed by comparing it to the ground truth using the intersection over union (IoU) metric. This metric measures the overlap between the ground truth and the predicted mask, and it is a dimensionless value that can also be employed for segmentation mask predictions. Typically, a threshold of 0.5 is utilized to differentiate between valid and invalid detections [38]. True positives (*TPs*), false positives (*FPs*), and false negatives (*FNs*) were determined based on the criteria outlined in Table 7. Precision and recall metrics were computed using Equations (1) and (2), respectively:

$$Precision\ (P) = \frac{TP}{TP + FP} \tag{1}$$

$$Recall\ (R) = \frac{TP}{TP + FN} \tag{2}$$

The *AP* for evaluation of the PCB defect detection was calculated from the graph plotted between precision and recall values, as shown in Equation (3):

$$AP = \sum_{k=0}^{n-1} [R(k) - R(k-1)] \cdot P(k) \tag{3}$$

where *n* is the number of defects identified.

Table 7. Definition of true positive (*TP*), false positive (*FP*), and false negative (*FN*).

Label	Description	Representation
TP	The object is there, and the model detects it with an IoU > 0.5.	 — Ground Truth — Prediction
FP	The object is there, but the model detects it with an IoU ≤ 0.5.	 — Ground Truth — Prediction
	The object is not there, and the model detects one.	 — Ground Truth — Prediction
FN	The object is there, and the model does not detect it.	 — Ground Truth — Prediction
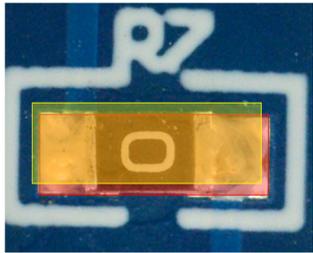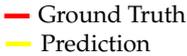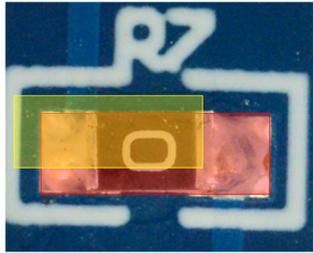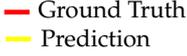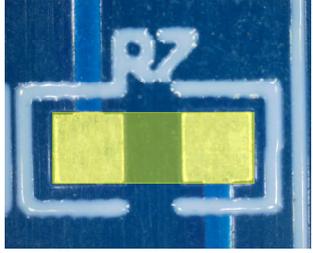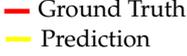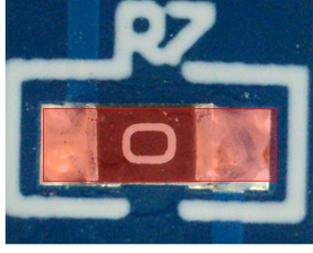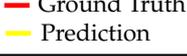
The COCO (common objects in context) detection metrics provide a way to evaluate the performance of object detection algorithms. These metrics include average precision (AP) and average recall (AR) values, which can be calculated using different intersection over union (IoU) thresholds, different numbers of detections, and different object scales.

Table 8 gives a breakdown of these metrics for various scenarios. The table includes different IoU thresholds, such as 0.5, 0.75, or higher, which determine the level of overlap required between the predicted and ground truth bounding boxes for a detection to be considered correct. It also includes results for different object scales, which refer to the dimensions of the identified objects, enabling an assessment of how well the algorithm performs across different object sizes.

Table 8. Performance metrics used (area expressed in pixels).

Average Precision (AP)	
AP	Percentage of AP at IoU = 0.50:0.05:0.95
AP $^{IoU = 0.50}$	Percentage of AP at IoU = 0.50
AP $^{IoU = 0.75}$	Percentage of AP at IoU = 0.75
Average Precision Across Scales	
AP ^{small}	Percentage of AP for small objects: area < 32^2
AP ^{medium}	Percentage of AP for medium objects: 32^2 < area < 96^2
AP ^{large}	Percentage of AP for large objects: area > 96^2
AP ^{all}	Percentage of AP not considering the size of the detection

In object detection tasks, AP is calculated by plotting a precision–recall curve for each class and computing the area under the curve (AUC). AP measures how well the model identifies objects of a specific class and assigns them accurate bounding boxes.

However, mAP (i.e., mean average precision) extends this concept by averaging the AP values for all classes. This provides a single and comprehensive score that reflects the model's overall performance across all object classes. It gives a more holistic view of the model's ability to detect objects in various scenarios, accounting for precision and recall across multiple classes simultaneously.

In Detectron2, several models come pre-trained. These models have already learned from existing datasets, and their weights can be fine-tuned to custom datasets. This allows them to be readily usable for specific tasks. During the training phase, a multi-loss function is employed to evaluate how well the model adapts to new datasets. The loss function incorporates three components with varying weights (as indicated in Equation (4)): classification error (*loss_cls*), localization error (*loss_box_reg*), and segmentation error (*loss_mask*).

$$loss_{tot} = loss_{cls} + loss_{box_{reg}} + loss_{mask} \tag{4}$$

In particular, *loss_cls* (classification loss) measures the differences between the predicted class probabilities and the actual class labels for each object in the training data. It is used to train the model to classify objects into their respective categories correctly. *Loss_box_reg* (bounding box regression loss) measures the difference between the predicted bounding box coordinates and the ground truth bounding box coordinates for each object in the training data. It is used to train the model to localize objects within the image accurately. Then, *loss_mask* (mask prediction loss) measures the similarity between the predicted segmentation masks and the ground truth masks for each instance in the training data. It is used in instance segmentation tasks to train the model to segment objects from the background accurately. Finally, total_loss represents the sum of all individual loss components (*loss_cls*, *loss_box_reg*, and *loss_mask*) and is used as the objective function to optimize the model's parameters.

6. Training and Validation

6.1. Model Settings

This paper implemented two different Mask R-CNNs: the first trained on dataset_1 and the second trained on dataset_2. In both cases, the first step was to import Default-Trainer from the Detectron2 engine module. Then, the main parameters were defined as follows:

- Batch size = 128;
- Number of classes = 2 for dataset_1 (good and no_good);
- Number of classes = 4 for dataset_2 (good, exc_solder, poor_solder, spike);
- Learning rate = 0.0025;
- Max iteration parameter = 500.

The number of iterations depended on the dataset size and the complexity of the task. The model performance when trained on dataset_1 and dataset_2 can be evaluated through loss (computed on the training dataset) and AP values (calculated on the validation dataset).

6.2. Result for Dataset_1

In this section, results related to dataset_1 are shown. In Figure 16, it can be noted that during the model training, the loss function decreases and becomes stable with the increase in the number of iterations.
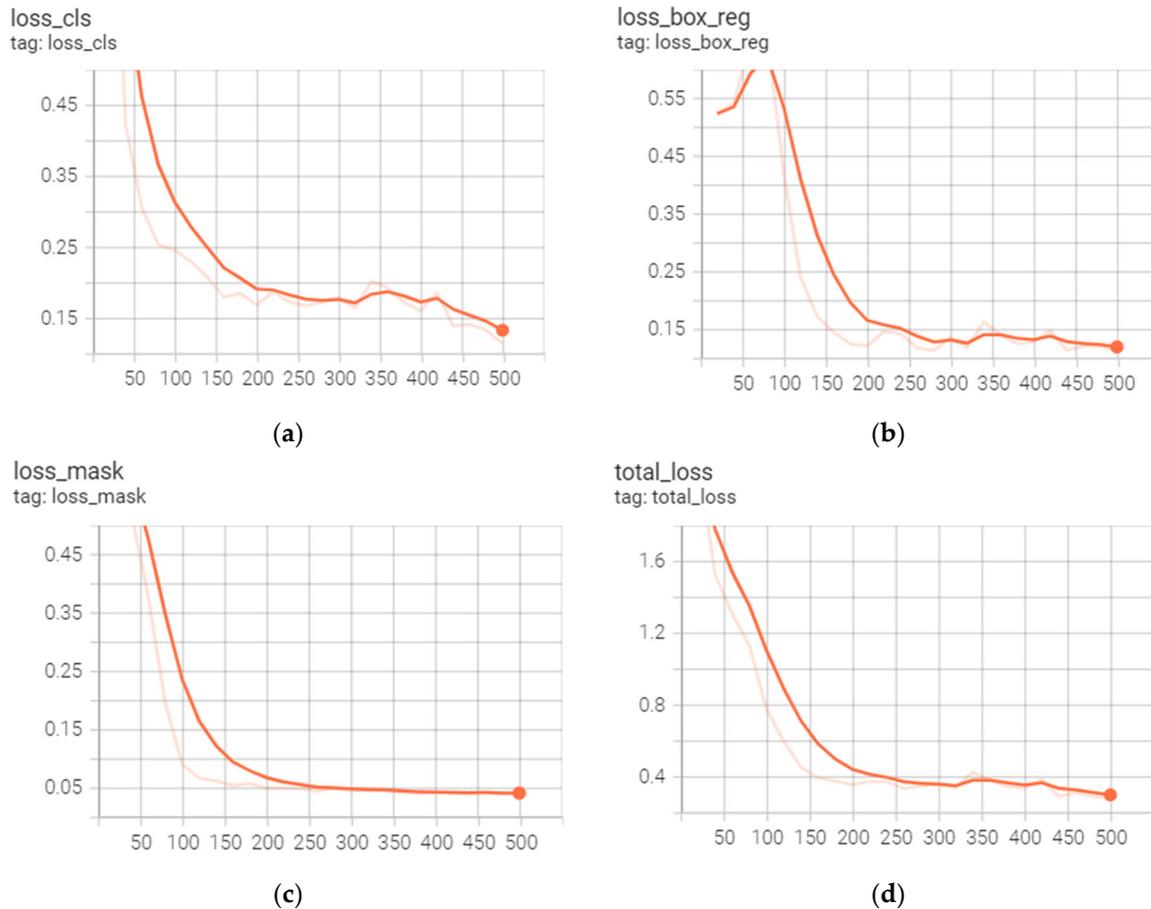
Figure 16. Loss function trends for classification (**a**), detection (**b**), segmentation (**c**), and total loss (**d**)—dataset_1.

The total loss function decreases to 0.282 (500 iterations). The loss function related to the classification task assumes a value close to zero (0.114) and the loss mask (0.041). The loss function referring to the detection (*loss_box_reg*) also tends to zero (reaching the value of 0.114), but it increases until the 80th iteration.

The model framework obtained good results regarding the precision of the detections and segmentations. The AP (100 max items detected, i.e., maxDets) for detection and segmentation yielded values of 62.11 and 68.57, respectively. Table 9 displays the evaluation metrics for the trained Mask R-CNN model (dataset_1).

Table 9. Evaluation of AP metric—dataset_1.

Metric	IoU	Area	maxDets	Value [%]
Detection				
mAP	@IoU = 0.50:0.05:0.95	All	100	62.11
mAP	@IoU = 0.50	All	100	71.05
mAP	@IoU = 0.75	All	100	71.05
mAP	@IoU = 0.50:0.05:0.95	Small (area < 32^2)	100	NaN
mAP	@IoU = 0.50:0.05:0.95	Medium (32^2 < area < 96^2)	100	NaN
mAP	@IoU = 0.50:0.05:0.95	Large (area > 96^2)	100	62.113
Segmentation				
mAP	@IoU = 0.50:0.05:0.95	All	100	68.57
mAP	@IoU = 0.50	All	100	71.05
mAP	@IoU = 0.75	All	100	71.05
mAP	@IoU = 0.50:0.05:0.95	Small (area < 32^2)	100	NaN
mAP	@IoU = 0.50:0.05:0.95	Medium (32^2 < area < 96^2)	100	NaN
mAP	@IoU = 0.50:0.05:0.95	Large (area > 96^2)	100	68.57

Figures 17–20 display four examples of results in detecting and segmenting the positioning of SMT components in the validation dataset (i.e., inference examples). The validation dataset consists of data not used to train the model. This subset of data enables evaluation of the model's performance on previously unseen data. The figures showcase bounding boxes, predicted shapes, and AP percentages.
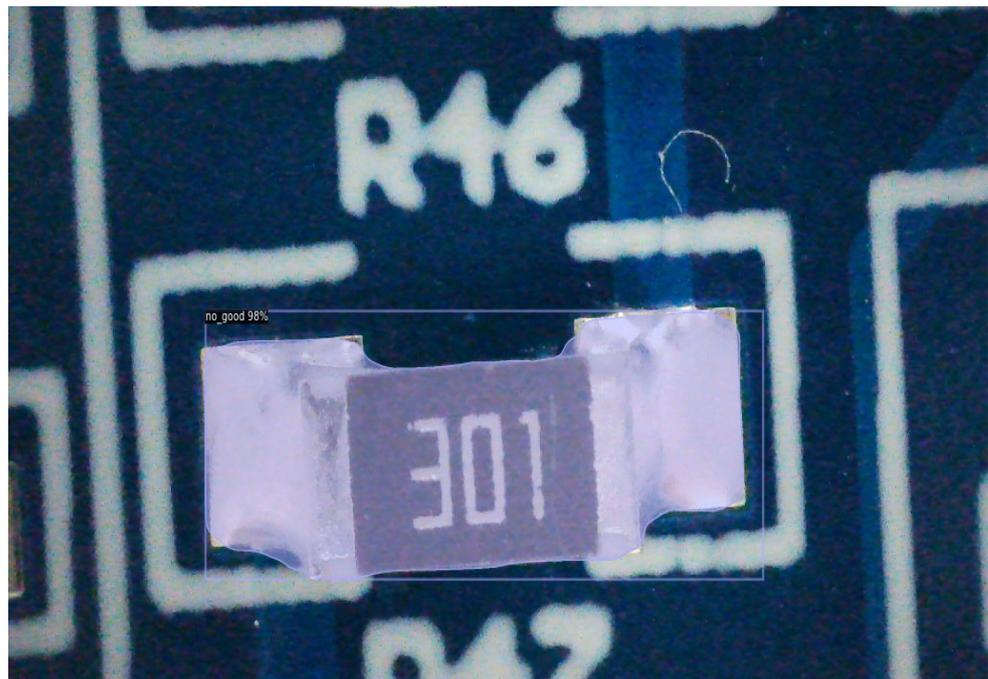
Figure 17. Inference example 1—dataset_1.

Figure 18. Inference example 2—dataset_1.
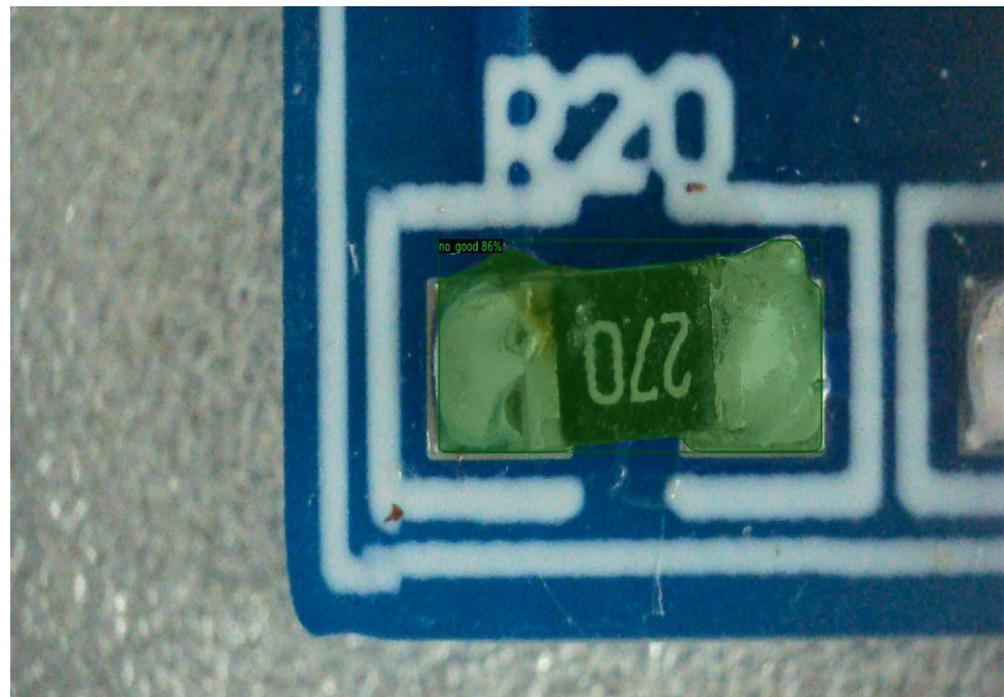
Figure 19. Inference example 3—dataset_1.

Figure 20. Inference example 4—dataset_1.

6.3. Result for Dataset_2

As conducted for the dataset referring to the position of the component on the PCBs, the performance metrics of the Mask R-CNN trained on dataset_2 (referring to the amount of solder material) are shown. The loss function trends related to classification (i), detection (ii), segmentation (iii), and total loss (iv) are illustrated in Figure 21. Differently from the previous case, the loss values are slightly higher. In particular, the loss function related to classification assumes a value of 0.226, the loss for detection is 0.295, the loss mask reaches a value of 0.138, and the total loss reaches a value of 0.745.

Concerning the precision of the detections and segmentations, the mAP (100 max items detected) values were equal to 42.70 and 48.30, respectively. All the evaluation metrics for the trained Mask R-CNN model's performance on dataset_2 are provided in Table 10.

These results also reflect the lower accuracy percentage of the predictions. Figures 22–27 show five examples of the detection and segmentation inferences for soldering defects performed on the validation dataset.

Table 10. Evaluation of AP metric—dataset 2.

Detection				
Metric	**IoU**	**Area**	**maxDets**	**Value [%]**
mAP	@IoU = 0.50:0.05:0.95	All	100	42.70
mAP	@IoU = 0.50	All	100	56.90
mAP	@IoU = 0.75	All	100	56.90
mAP	@IoU = 0.50:0.05:0.95	Small (area < 32^2)	100	NaN
mAP	@IoU = 0.50	Medium (32^2 < area < 96^2)	100	NaN
mAP	@IoU = 0.75	Large (area > 96^2)	100	42.70
Segmentation				
mAP	@IoU = 0.50:0.05:0.95	All	100	48.30
mAP	@IoU = 0.50	All	100	56.90
mAP	@IoU = 0.75	All	100	56.90
mAP	@IoU = 0.50:0.05:0.95	Small (area < 32^2)	100	NaN
mAP	@IoU = 0.50	Medium (32^2 < area < 96^2)	100	NaN
mAP	@IoU = 0.75	Large (area > 96^2)	100	48.30
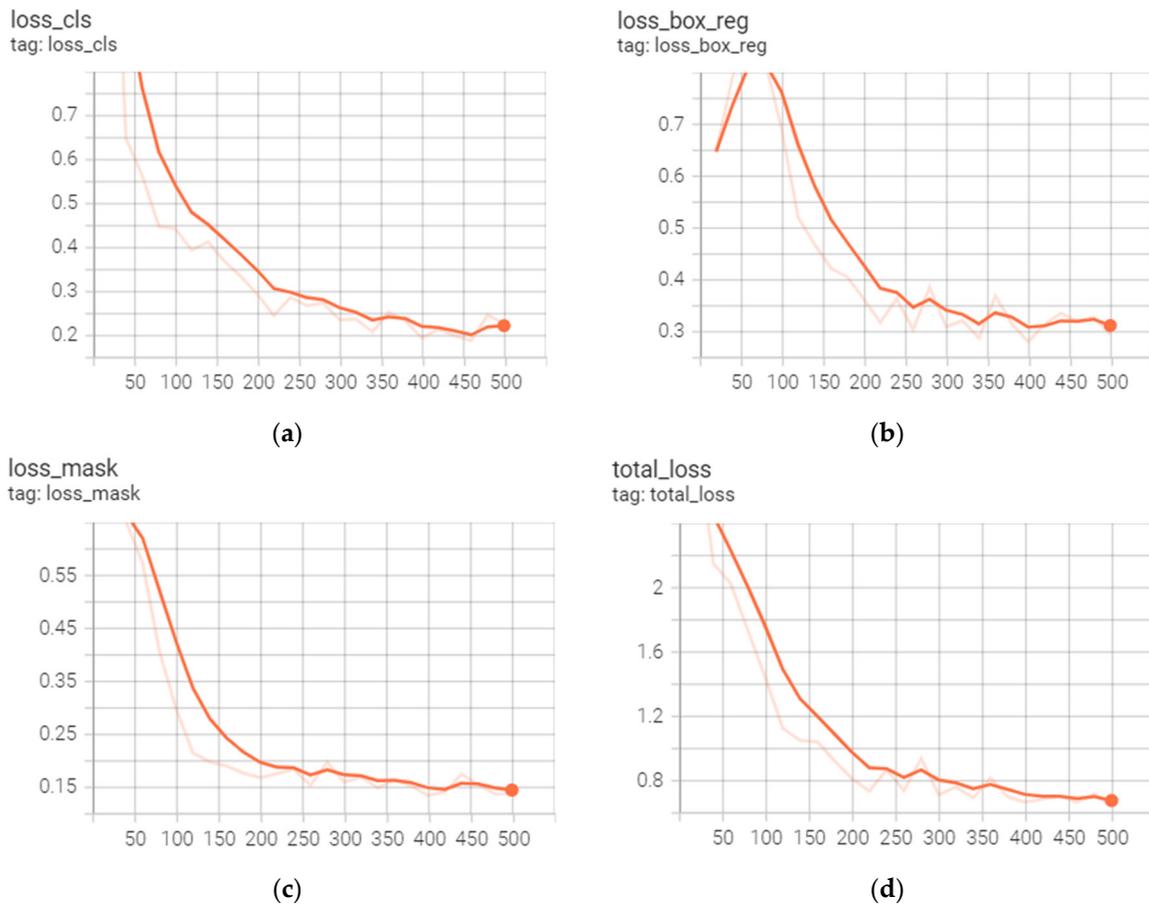
Figure 21. Loss function trends for classification (**a**), detection (**b**), segmentation (**c**), and total loss (**d**)—dataset_2.
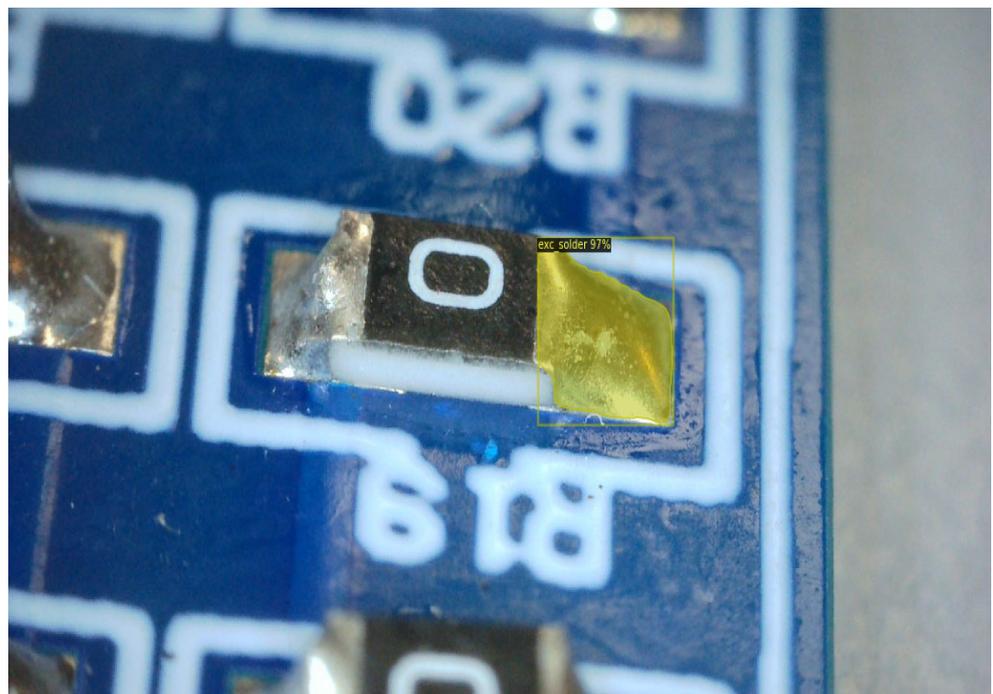
Figure 22. Inference example 1—dataset_2.
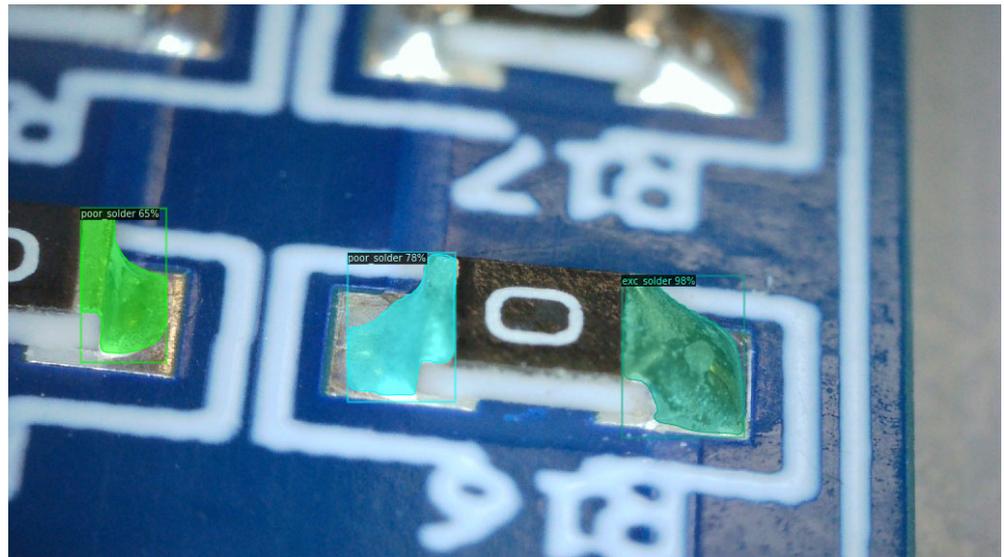
Figure 23. Inference example 2—dataset_2.
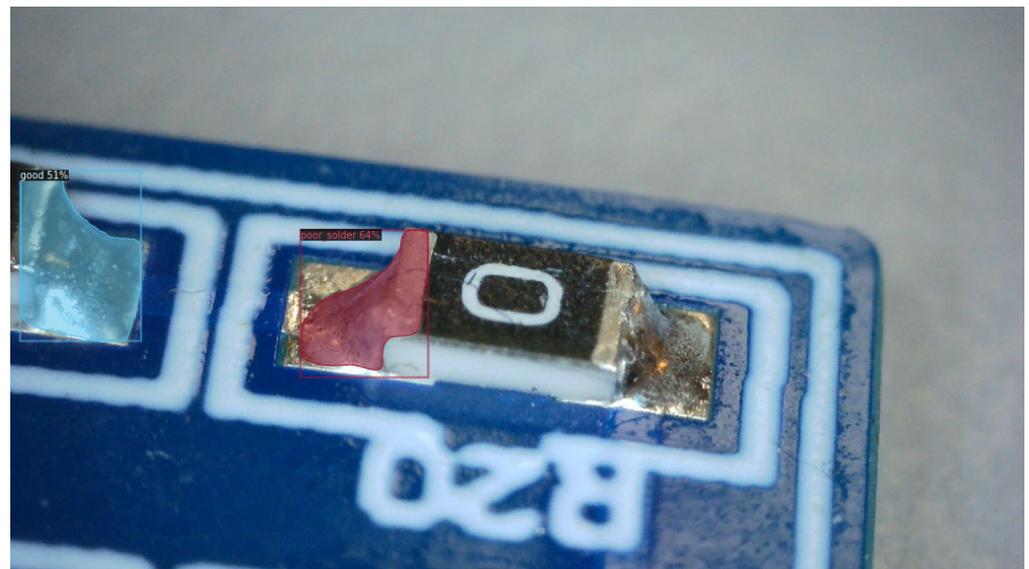
Figure 24. Inference example 3—dataset_2.

In this first image, one can appreciate the network difficulty in segmenting and classifying the left joint while achieving an accuracy value of 97% on the right joint, generating a perfectly adherent mask according to the labeling specifications. The associated label is also correct, as it represents excessive soldering material, as indicated by the convex profile of the joint's contour.

In the second image of the validation dataset, the network perfectly segments both the right and left joints, even if achieving different accuracy values (98% for the right joint and 78% for the left joint).

In the third image, the network fails to segment the correct joint. Additionally, the prediction accuracy for the left joint is not very high either (64%). However, the associated label still corresponds to the actual condition, and the generated mask successfully segments the reference area with great precision.
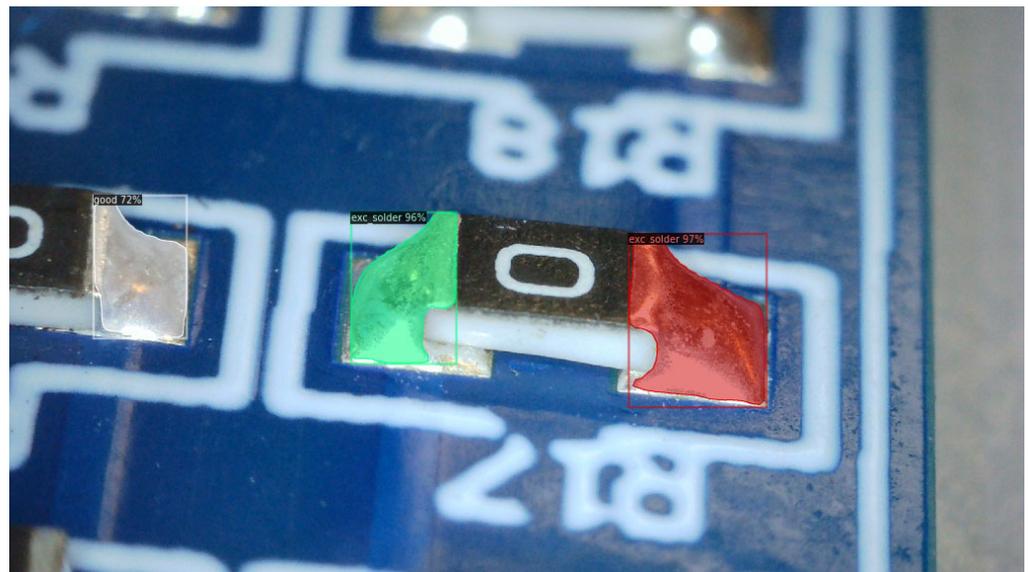
Figure 25. Inference example 4—dataset_2.
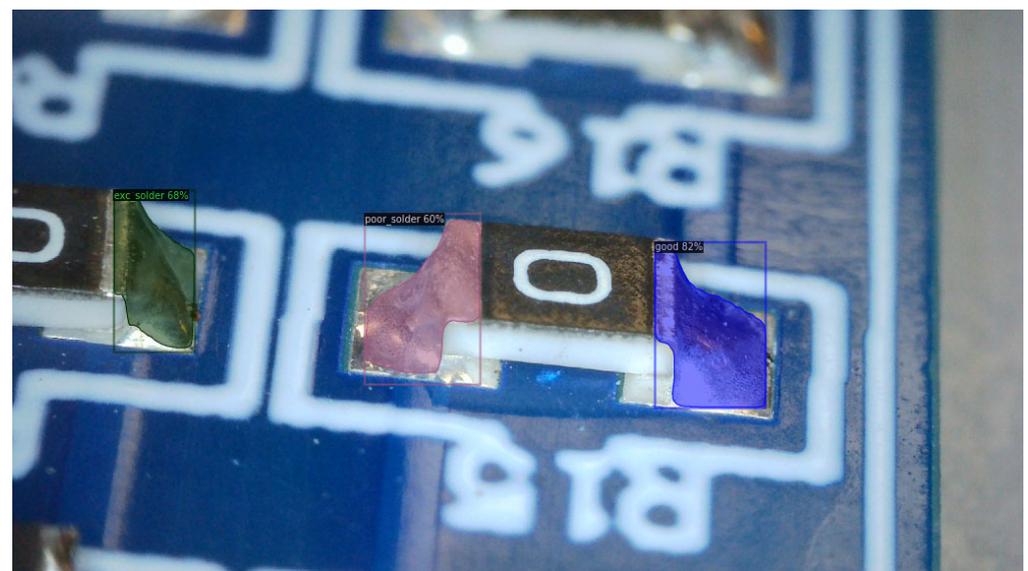
Figure 26. Inference example 5—dataset_2.

In the fourth image of the validation dataset, the best result in terms of accuracy is achieved. The generated masks fit perfectly with the actual solder joints, and the associated labels are as expected. The accuracy values for the left and right masks reach 96% and 97%, respectively.

In the fifth image of the validation set, the algorithm continues to provide correct results, although with lower accuracy values. Specifically, 60% is achieved for the poor solder and 82% for the properly executed joint. The generated masks also appear to be appropriately generated.

Further, in the last image example of the validation set, the algorithm continues to provide correct results, even with lower accuracy values. Specifically, 86% is achieved for the spike and 84% for the properly executed joint. The generated masks also appear to be appropriately generated.
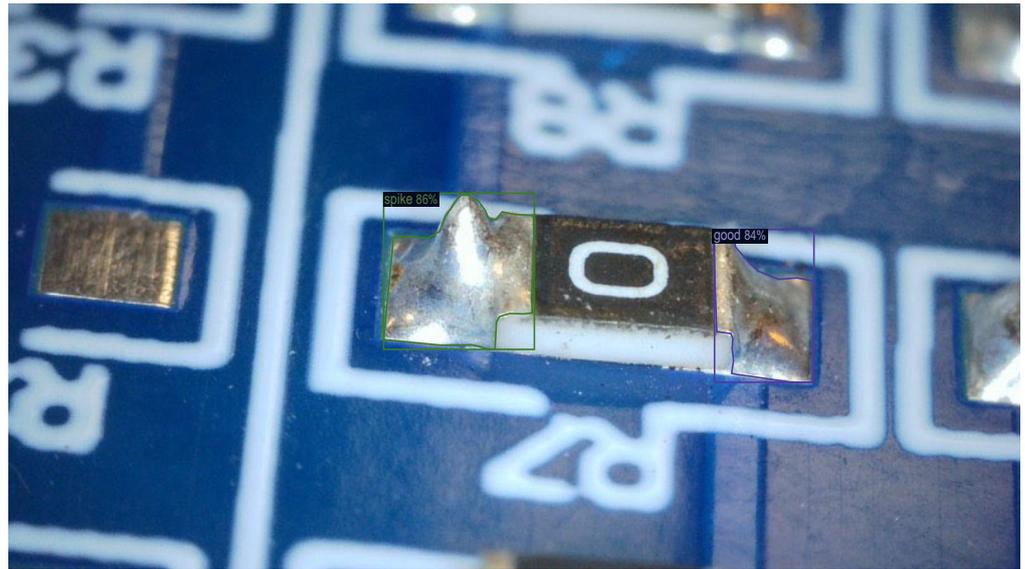
Figure 27. Inference example 6—dataset_2.

6.4. Result Discussion

As described before, dataset_1 includes top view images related to SMT positioning, while dataset_2 contains 45-degree view images related to the quantity of soldering material on joints. The obtained results for the defined datasets (dataset_1 and dataset_2) can be summarized with the following considerations:

- The algorithm better manages the top view images (dataset_1). In this case, the metrics highlight the robustness of the model. Indeed, for dataset_1, the total loss function decreased to 0.282, while in the 45-degree view (dataset_2), the total loss function was 0.745.
- The better performance for dataset_1 was also confirmed by other metrics (loss to the classification task, loss mask, loss to the detection, mAP for detection, and mAP for segmentation).
- For dataset_1, the algorithm managed only two classes (good and no_good); meanwhile, in the case of dataset_2, the number of classes was four (good, exc_solder, poor_solder, spike).
- The algorithm performances are interesting for the two datasets. However, the results highlight some limits to managing the soldering defects in the 45-degree images (dataset_2). Indeed, the algorithm had more difficulty detecting especially the class poor_solder. For the other classes (good, exc_solder, and spike), the algorithm, on average, gave better performances, but with rare cases in which the inference was lower than 70%. This behavior probably depends on boundaries that are not identifiable clearly due to the defects' characteristics and the point of view of the image.
- Further studies should be carried out for the exc_solder and poor_solder classes to overcome the aforementioned limitations.

Nevertheless, the obtained results are surely interesting and introduce several elements of innovation for the context of the application (i.e., PCB manufacturing for the aerospace sector), and the presented solution is a candidate to be a valid support to technicians for quality control in compliance with the stringent standards. Indeed, these processes are currently performed manually and, therefore, are subject to human errors.

To improve the performance of the algorithm, specifically in the exc_solder and poor_solder classes, future studies will be conducted on the following:

- Updating the SolDef_AI dataset with a new experimental campaign to acquire new images with different points of view: 15 degrees, 30 degrees, 45 degrees, 60 degrees,

and 75 degrees. Indeed, the SolDef_AI dataset represents a dynamic open source dataset with scheduled updates.

- Optimizing the training phase with improvements in the labeling strategies, dimension of the training sub-datasets, and model settings.

Future developments will aim to improve the performance of the model in order to perform its integration within an embedded system for defect detection to be used in real industrial applications.

7. Conclusions

This paper presents a new open source dataset (SolDef_AI) for the implementation of an innovative methodology for PCB defect detection exploiting the Mask R-CNN algorithm. SolDef_AI aims to overcome the barriers that characterize the development of tools for defect recognition in the case of PCBs for aerospace applications. Indeed, in this case, the soldering of micro/meso electronic components must respect specific standards such as ECSS-Q-ST-70-38C.

This research required a preliminary activity to define the components of interest (i.e., SMT types C0805, R0603, R0805, and R1206) and the related PCB board. After this, the system of acquisition was defined and configured. Each component was acquired from three points of view (top, 45-degree, and axonometric views).

Starting from the dataset acquisition, a labeling campaign was carried out to create the training sub-dataset. The labeling strategies were defined considering the constraints defined by the ECSS-Q-ST-70-38C standard.

The dataset was used to train a Mask R-CNN model. Indeed, it stands out in pixel-level precision and accuracy, for instance, in segmentation. Its ability to deliver detailed segmentation masks enhances its effectiveness in scenarios where a nuanced understanding of object boundaries is critical.

The results of the work allow to introduce some innovative elements, such as the development and introduction of a new open source dataset in the scientific community for this specific context. Indeed, the new dataset will allow the testing of new solutions for aerospace applications, considering the requirements defined by the stringent standards. Moreover, due to the training with the latest open source dataset (SolDef_AI), the Mask R-CNN allowed the recognition of the main PCB defects of interest for aerospace applications with a specific focus on soldering STM components. Indeed, the Mask R-CNN provides superior pixel-level precision and accuracy compared to alternative models such as YOLO (you only look once) and SSD (single shot multi-box detector). This is particularly crucial in scenarios where precise boundary recognition, such as in the case of soldering materials or SMT components, is required in the segmentation process. On the other hand, YOLO and SSD might surpass models such as Mask R-CNN concerning speed and applications requiring real-time processing owing to their single-shot approach, which analyzes the entire image in a single iteration.

The performance evaluation of the Mask R-CNN models trained on dataset_1 and dataset_2 revealed notable results. For dataset_1, the loss function trends indicated a steady decrease with increasing iterations, achieving a total loss of 0.282. Precision metrics, including mean average precision (mAP), for detection and segmentation demonstrated a strong performance with values of 62.11 and 68.57, respectively. Conversely, for dataset_2, slightly higher loss values were observed, with total loss reaching 0.745. Despite this, the mAP values for detection and segmentation remained respectable at 42.70 and 48.30, respectively. These findings suggest that the Mask R-CNN models effectively captured and classified components on PCBs, showcasing promising capabilities for defect detection and segmentation tasks in electronic manufacturing processes.

This study lays the foundation for the implementation of an embedded solution to monitor soldering defects in the manufacturing of PCBs for aerospace applications.

The implemented model highlights notable results in detecting assembly defects for the two datasets. Indeed, the obtained metrics show the robustness of the model in

recognizing assembly defects. However, there are some limits to managing the soldering defects in the 45-degree images (dataset_2). To overcome the observed limitations, future developments will be focused on improving the performance of the algorithm, updating the SolDef_AI dataset with new images, and carrying out studies for the optimization of the training phase.

In real industrial scenarios, the possibility for technicians to have valid support to detect whether solder joints comply with aerospace standards or not is already an important enhancement considering that these processes are often performed manually based on the knowledge of the operators. Future developments will allow to improve the performance of the model and, therefore, increase the support that this solution can provide to the technicians. The goal, indeed, is to develop an embedded system that consists of a vision camera with software for defect detection.

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

References

1. Ackermann, M.; Iren, D.; Wesselmecking, S.; Shetty, D.; Krupp, U. Automated segmentation of martensite-austenite islands in bainitic steel. *Mater. Charact.* **2022**, *191*, 112091. [CrossRef]
2. Karem, M. Reviewing Mask R-CNN: An In-depth Analysis of Models and Applications. *EasyChair* **2024**, 11838.
3. Chen, X.; Wu, Y.; He, X.; Ming, W. A Comprehensive Review of Deep Learning-Based PCB Defect Detection. *IEEE Access* **2023**, *11*, 139017–139038. [CrossRef]
4. Park, J.-H.; Kim, Y.-S.; Seo, H.; Cho, Y.-J. Analysis of Training Deep Learning Models for PCB Defect Detection. *Sensors* **2023**, *23*, 2766. [CrossRef] [PubMed]
5. Chen, I.-C.; Hwang, R.-C.; Huang, H.-C. PCB Defect Detection Based on Deep Learning Algorithm. *Processes* **2023**, *11*, 775. [CrossRef]
6. Lian, J.; Wang, L.; Liu, T.; Ding, X.; Yu, Z. Automatic visual inspection for printed circuit board via novel Mask R-CNN in smart city applications. *Sustain. Energy Technol. Assess.* **2021**, *44*, 101032. [CrossRef]
7. Liu, Y.; Wu, H. Automatic Solder Defect Detection in Electronic Components Using Transformer Architecture. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2023**, *14*, 166–175. [CrossRef]
8. Xin, H.; Chen, Z.; Wang, B. PCB Electronic Component Defect Detection Method based on Improved YOLOv4 Algorithm. *J. Phys. Conf. Ser.* **2021**, *1827*, 012167. [CrossRef]
9. Wu, X.; Ge, Y.; Zhang, Q.; Zhang, D. PCB defect detection using deep learning methods. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021.
10. Mujeeb, A.; Dai, W.; Erdt, M.; Sourin, A. One class based feature learning approach for defect detection using deep autoencoders. *Adv. Eng. Inform.* **2019**, *42*, 100933. [CrossRef]
11. Sezer, A.; Altan, A. Detection of solder paste defects with an optimization-based deep learning model using image processing techniques. *Solder. Surf. Mt. Technol.* **2021**, *33*, 291–298. [CrossRef]
12. Wang, H.; Xie, J.; Xu, X.; Zheng, Z. Few-Shot PCB Surface Defect Detection Based on Feature Enhancement and Multi-Scale Fusion. *IEEE Access* **2022**, *10*, 129911–129924. [CrossRef]
13. Kim, J.; Ko, J.; Choi, H.; Kim, H. Printed Circuit Board Defect Detection Using Deep Learning via A Skip-Connected Convolutional Autoencoder. *Sensors* **2021**, *21*, 4968. [CrossRef] [PubMed]
14. Benedek, C.; Krammer, O.; Janoczki, M.; Jakab, L. Solder Paste Scooping Detection by Multilevel Visual Inspection of Printed Circuit Boards. *IEEE Trans. Ind. Electron.* **2012**, *60*, 2318–2331. [CrossRef]

15. Öztürk, Ş.; Akdemir, B. Detection of pcb soldering defects using template based image processing method. *Int. J. Intell. Syst. Appl. Eng.* **2017**, *4*, 269–273.

16. Vakili, E.; Karimian, G.; Shoaran, M.; Yadipour, R.; Sobhi, J. Valid-IoU: An improved IoU-based loss function and its application to detection of defects on printed circuit boards. *Res. Sq.* **2023**. *preprint*. [CrossRef]

17. Bártová, B.; Bína, V. A Novel Data Mining Approach for Defect Detection in the Printed Circuit Board Manufacturing Process. *Eng. Manag. Prod. Serv.* **2022**, *14*, 13–25. [CrossRef]

18. Ding, R.; Dai, L.; Li, G.; Liu, H. TDD-net: A tiny defect detection network for printed circuit boards. *CAAI Trans. Intell. Technol.* **2019**, *4*, 110–116. [CrossRef]

19. Liao, X.; Lv, S.; Li, D.; Luo, Y.; Zhu, Z.; Jiang, C. YOLOv4-MN3 for PCB Surface Defect Detection. *Appl. Sci.* **2021**, *11*, 11701. [CrossRef]

20. Park, J.-M.; Yoo, Y.-H.; Kim, U.-H.; Lee, D.; Kim, J.-H. D^{3}PointNet: Dual-Level Defect Detection PointNet for Solder Paste Printer in Surface Mount Technology. *IEEE Access* **2020**, *8*, 140310–140322. [CrossRef]

21. Wan, Y.; Gao, L.; Li, X.; Gao, Y. Semi-Supervised Defect Detection Method with Data-Expanding Strategy for PCB Quality Inspection. *Sensors* **2022**, *22*, 7971. [CrossRef] [PubMed]

22. Zhang, G.; Cao, Y. A novel pcb defect detection method based on digital image processing. *J. Phys. Conf. Ser.* **2023**, *2562*, 012030. [CrossRef]

23. An, K.; Zhang, Y. LPViT: A Transformer Based Model for PCB Image Classification and Defect Detection. *IEEE Access* **2022**, *10*, 42542–42553. [CrossRef]

24. Li, Q.; Zheng, Q.; Jiang, S.; Hu, N.; Liu, Z. An improved YOLOv5-based model for automatic PCB defect detection. *J. Physics Conf. Ser.* **2024**, *2708*, 012017. [CrossRef]

25. Tang, J.; Liu, S.; Zhao, D.; Tang, L.; Zou, W.; Zheng, B. PCB-YOLO: An Improved Detection Algorithm of PCB Surface Defects Based on YOLOv5. *Sustainability* **2023**, *15*, 5963. [CrossRef]

26. Calabrese, M.; Agnusdei, L.; Fontana, G.; Papadia, G.; Del Prete, A. Application of Mask R-CNN for Defect Detection in Printed Circuit Board manufacturing. *SN Appl. Sci.* **2024**. *preprint*.

27. Lu, H.; Mehta, D.; Paradis, O.; Asadizanjani, N.; Tehranipoor, M.; Woodard, D.L. FICS-PCB: A Multi-Modal Image Dataset for Automated Printed Circuit Board Visual Inspection. *Cryptol. Eprint Arch.* **2020**. *preprint*.

28. Tang, S.; He, F.; Huang, X.; Yang, J. Online pcb defect detector on a new pcb defect dataset. *arXiv* **2019**, arXiv:1902.06197.

29. Huang, W.; Wei, P. A pcb dataset for defects detection and classification. *arXiv* **2019**, arXiv:1901.08204.

30. Mahalingam, G.; Gay, K.M.; Ricanek, K. PCB-METAL: A PCB Image Dataset for Advanced Computer Vision Machine Learning Component Analysis. In Proceedings of the 2019 16th International Conference on Machine Vision Applications (MVA), Tokyo, Japan, 27–31 May 2019; pp. 1–5. [CrossRef]

31. Pramerdorfer, C.; Kampel, M. A dataset for computer-vision-based PCB analysis. In Proceedings of the 2015 14th IAPR International Conference on Machine Vision Applications (MVA), Tokyo, Japan, 18–22 May 2015; pp. 378–381. [CrossRef]

32. Fontana, G.; Ruggeri, S.; Fassi, I.; Legnani, G. Flexible Vision Based Control for Micro-Factories. In Proceedings of the ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Portland, OR, USA, 4–7 August 2013. [CrossRef]

33. Ruggeri, S.; Fontana, G.; Fassi, I. Micro-assembly. In *Micro-Manufacturing Technologies and Their Applications*; Fassi, I., Shipley, D., Eds.; Springer Tracts in Mechanical Engineering; Springer: Cham, Switzerland, 2017. [CrossRef]

34. Basile, V.; Modica, F.; Fontana, G.; Ruggeri, S.; Fassi, I. Improvements in Accuracy of Fused Deposition Modeling Via Integration of Low-Cost On-Board Vision Systems. *J. Micro Nano-Manuf.* **2020**, *8*, 010905. [CrossRef]

35. Ruggeri, S.; Fontana, G.; Legnani, G.; Fassi, I. Performance Indices for the Evaluation of Microgrippers Precision in Grasping and Releasing Phases. *Int. J. Precis. Eng. Manuf.* **2019**, *20*, 2141–2153. [CrossRef]

36. Kitada, T.; Seki, Y. Mounting Technique of 0402-Sized Surface-Mount Device (SMD) on FPC. *Fujikura Tech. Rev.* **2011**, 29.

37. Sezer, A.; Altan, A. Optimization of Deep Learning Model Parameters in Classification of Solder Paste Defects. In Proceedings of the 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 11–13 June 2021; pp. 1–6.

38. Rastegari, M.; Ordonez, V.; Redmon, J.; Farhadi, A. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*; Springer International Publishing: Cham, Switzerland, 2016; pp. 525–542. [CrossRef]