
Developing a YOLOv3-Based Neural Network to Enable Digital Cameras to Detect Features on the Reflective Surface of Objects in Robotic Applications: A Low-Computational Approach with a New Public Dataset

[Albin Bajrami](#)* and [Matteo Claudio Palpaceli](#)

Posted Date: 5 March 2024

doi: 10.20944/preprints202403.0221.v1

Keywords: Feature Detection; Object Reflective Surfaces; YOLOv3 Neural Network; Transfer Learning; Digital Cameras; Collaborative Robots; Machine Learning in Manufacturing; Automation in Polishing; Industrial AI Applications; GPU; CUDA and cuDNN



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Developing a YOLOv3-Based Neural Network to Enable Digital Cameras to Detect Features on the Reflective Surface of Objects in Robotic Applications: A Low-Computational Approach with a New Public Dataset

Albin Bajrami ^{*,†}  and Matteo Claudio Palpacelli

Università Politecnica delle Marche; m.c.palpacelli@staff.univpm.it

* Correspondence: a.bajrami@pm.univpm.it

† The Italian Doctorate in Robotics and Intelligent Machines.

Abstract: This study explores the integration of Artificial Intelligence (AI) and collaborative robotics in industrial surface finishing, focusing on feature detection on object reflective surfaces using neural networks in computationally constrained environments. Central to this investigation is the utilization of cameras considered as sensors that can capture detailed images of reflective surfaces. These images are then analyzed by a custom neural network, demonstrating the feasibility of feature detection with non-powerful computers. The public dataset for recognizing light reflection defects, available in the SPADD repository, supports this approach. Employing open-source tools like Darknet YOLOv3 and Yolo_mark, this method proves particularly advantageous for small and medium-sized enterprises (SMEs) due to its low cost and maintenance requirements. This research highlights the challenges of AI and collaborative robotics in industrial applications, especially in adapting to varying conditions and the need for continuous training with new data coming from several sensors. Looking ahead, plans are in place to integrate this system with cobot-assisted polishing tasks, using cameras to enhance manufacturing efficiency and quality. This work contributes to the broader understanding of practical applications of AI in industrial automation, surface finishing processes, and the effective use of sensor technology.

Keywords: feature detection; object reflective surfaces; YOLOv3 neural network; transfer learning; digital cameras; collaborative robots; machine learning in manufacturing; automation in polishing; industrial AI applications; GPU; CUDA and cuDNN

1. Introduction

Technological progress in the fields of Artificial Intelligence (AI) and Robotics is radically changing the industrial landscape leading to significant innovations in numerous sectors. In particular, the use of AI and collaborative robots in manufacturing is opening up new possibilities for the efficiency and quality of production processes. Surface finishing, a critical process in many industrial applications, is increasingly making use of these technologies to improve accuracy, reduce processing times, and achieve higher quality results through enhanced sensing capabilities.

1.1. Collaborative Robotics

Collaborative robots, also known as 'cobots', represent a major revolution in the field of industrial automation. These devices are changing the traditional paradigm for the use of robots in companies. In the past, robots were mainly used in isolated areas where they worked completely autonomously and separately from humans. Their presence was limited to robotic cells where they performed repetitive tasks without direct human interaction.

Today, cobots are equipped with advanced sensors that allow them to safely 'sense' and interact with the world around them. These on-board sensors, which include force sensors and vision systems, as well as artificial intelligence algorithms, allow cobots to dynamically adapt to the different tasks the robot has to perform. Recent developments in robotic systems driven by AI algorithms have

significantly improved their adaptability and flexibility in various production processes. Garcia et al. [1], show, for example, how the integration of artificial intelligence into collaborative robots is crucial for optimising flexible, multi-variant manufacturing. By using machine learning and AI models, these robots can dynamically adapt to changes in the assembly process, significantly reducing the need for manual reprogramming. This approach not only increases production efficiency, but also opens up new perspectives for innovation in industrial automation.

1.2. Artificial Intelligence and Its Importance

AI is playing a crucial role in a variety of disciplines and fields of knowledge, revolutionising the way we approach and solve complex problems in different areas. To name just a few examples, Sowa et al. in [2] emphasise the importance of synergetic collaboration between human workers and AI systems in management tasks. The study underlines how positively employees view the use of such intelligent systems, but also express concerns about full automation.

In the field of mechatronics, Bohušík et al. in [3] use artificial intelligence algorithms for the precise control of three motors for end-effector control of a parallel kinematic mechanism and show how the neural networks used make it possible to correct mechanical inaccuracies and improve the performance of the mechanical system.

In [4], Bhardwaj et al. conduct a comprehensive analysis of the use of artificial intelligence (AI) in the life sciences, focusing on its impact on improving quality of life. This article examines how AI is being used in critical areas such as accurate medical diagnosis, pharmaceutical research, patient-tailored medicine and gene editing. In agriculture, AI is helping to reduce waste and increase crop yields, playing a key role in precision agriculture. The article also highlights the crucial role of AI in reducing operating costs and increasing production in specialised areas such as bio-industrial enzymes, underlining the critical importance of these technologies for sustainable progress.

In the field of space exploration, the combination of robotics and artificial intelligence is an important pillar for assisting with dangerous tasks that traditionally require human intervention. According to Nanjangud et al. in [5], these advanced technologies are proving critical in three key mission areas. The first, called Passive On-Orbit Servicing, concerns operations such as satellite inspection. The second, Active Debris Removal and Active On-Orbit Servicing, involves debris removal and other forms of technical support. Finally, on-orbit assembly, which includes the assembly and installation of structures in orbit. The article highlights that Robotic and Autonomous Systems (RAS) are playing an increasingly important role in these scenarios, reducing the need for direct human intervention in high-risk situations. Thanks to advanced sensor technology, RAS can perceive their environment and use artificial intelligence algorithms to convert the collected data into autonomous decisions. This enables robots to perform tasks fully or partially autonomously, increasing the efficiency and safety of space operations.

Several other publications utilize AI for various purposes. An example is provided by Carbonari et al. in [6], who apply artificial intelligence to vision in an informative context. They use the collaborative robot Yumi to play the game *Connect Four*, thereby demonstrating the effectiveness of AI in a playful activity that requires both visual perception and decision-making capabilities.

In the manufacturing sector, AI is significantly changing the possibilities of autonomous perception and decision-making in post-processing operations such as polishing and machining. The use of neural networks opens up new avenues for the precise recognition and classification of objects and, particularly in the industrial sector, the detection of objects to be gripped or surface defects to be recognised and reworked where possible. This innovation enables robots to adapt their actions to the specific characteristics of the materials to be handled and the task to be performed. This advance not only increases the efficiency and consistency of industrial processes, but also allows for greater customization and quality of the end product, as well as easier training. Ease of programming is a key issue for industrial robotics. Simple programming is a key issue for industrial robotics. In a recent publication, Kaczmarek et al. experimented in [7] with the use of a graphics tablet to simplify the pro-

programming of industrial robots. For this purpose, an application was implemented in C# that integrates an intuitive user interface and a digital twin for testing and optimising trajectories. Task-oriented programming and not the individual operations to be carried out are crucial for the integration of robotics in small and medium-sized enterprises (SMEs) as well.

Literary revisions on the integration of AI in manufacturing, such as that of Pandiyan et al. [8], provide a detailed analysis on the advancements and applications of AI techniques in abrasive finishing processes. These studies examine how AI can be employed to improve accuracy, efficiency, and monitoring in material processing, highlighting the importance of smart manufacturing (SM) and optimized process management.

Concurrently, the review by Kim et al. [9] explores the use of machine learning in SM processes, highlighting how these technologies can enhance sustainability and reduce environmental impact in processing industries. SM is a set of manufacturing practices that use networked data and information and communication technologies (ICTs) for governing manufacturing operations [10] [11]. The survey by Luo et al. [12] conducts an overview of the technologies used for surface defect recognition, with a perspective on quality control, analyzing about 120 publications produced in the last twenty years for three typical flat steel products (conglomerate slabs, hot and cold rolled steel strips).

The literary review so far reported aims to show how the use of artificial intelligence is becoming an increasingly determining phenomenon in all sectors where human decision-making is required and not only.

1.3. The Research Focus

This research focuses on bridging the gap in the field of artificial intelligence and robotics, specifically in the recognition of object features on their reflective surfaces through the use of neural networks, especially in environments with limited computational capacity. The aim is to develop an efficient and viable approach to improve surface finishing processes in the industry, utilizing the power of AI in a more accessible and cost-effective computational context. The main problems addressed in this research are:

- *Gap in Reflective Surface Recognition:* Despite advances in AI and robotics, there remains a gap in the effective recognition of reflective objects through neural networks, particularly in environments with limited computational resources;
- *Development of a Public Dataset:* Creation of an evolving public dataset for the recognition of surface defects related to light reflection;
- *Defect Detection in Low-Computational-Capacity Environments:* Demonstration that the detection of such defects can be executed even in environments with less powerful hardware;
- *Use of FOS Tools for Neural Network:* Utilization of Free and Open Source (FOS) tools to simplify the creation of the neural network and its publication for the scientific community;
- *Network Development Methodology:* Description of the adopted methodology, including the creation of the network on less performing PCs and the training of weights on more powerful devices by using the same procedure presented in [13];
- *Creation and Publication of Weights for Transfer Learning:* Development and sharing of weights to facilitate transfer learning in the community;
- *Publication of Results:* Dissemination of the results obtained from the study, contributing to collective knowledge in the field.

1.4. Structure of the Paper

The paper is structured as follows: *Section Two* delves into the framework within which this project is placed, providing key insights necessary to understand the focus of this paper. It discusses the challenges associated with polishing tasks and the need for a robotic framework to automate post-production processes. *In Section Three*, is presented a comprehensive overview of the system. This includes a detailed description of the system components and configurations. The chapter explores the process of image acquisition for network training and analyzes the system's performance, covering

hardware aspects and optimization techniques such as CUDA and cuDNN. *Section Four* is dedicated to describing the dataset developed for this research. It details the creation process and highlights the current limitations of the dataset. *Section Five* delves into the design of the custom Darknet YOLOv3 neural network. The discussion focuses on the specific roles of various layers, including convolutional and max pooling layers, within the network, providing insights into the network architecture and functionality. Finally, *Section Six* summarizes the results and conclusions of the study. It covers the development and training of the dataset and network, examining the network performance in limited computational environments. The chapter discusses the training methodology and data augmentation techniques employed and examines the implications of this research in the industrial context. The chapter concludes by looking forward to future integrations with collaborative robot polishing tasks, highlighting the practical applications of the research in industrial settings.

2. Flexible Framework for Polishing

In the initial paragraph, we discussed post-processing, highlighting the use of Cobots and AI in this context. This paragraph aims to more specifically define the aspects of post-processing addressed in the article and introduces the robotic framework designed for solving the problem of easy way to polish complex surface by cobots.

2.1. The Problems Related to Polishing Task

Post-processing represents the final stage of the manufacturing process and includes a series of steps that a product must undergo before being used for its final purpose. It is a general term that encompasses various stages necessary before a product can be employed. The objective of post-processing is to remove the undesired properties accumulated in the final product during the manufacturing process. By 'manufacturing process', we mean the steps and actions required to transform raw materials or semi-finished products into finished or semi-finished products. There are various types of manufacturing processes, including Molding, Forming, Mechanical Processing, Assembly, Cutting, and the recent 3D Printing.

Generally, the post-processing phase can include, but is not limited to, heat treatment, UV curing, support removal, cleaning and depowdering, machining, coating or infiltration, surface finishing processes, inspection, and dyeing. Nowadays, in additive manufacturing, post-processing is a crucial aspect, as it helps to address issues such as poor surface quality and inadequate mechanical properties as reported by Peng et al. [14]. This is particularly evident in the need for post-process machining of additive manufactured parts to improve surface finish and geometric accuracy as noted by Lane et al. [15]. Mishra et al. [16] further emphasizes the importance of post-processing techniques, categorizing them into methods for support material removal, surface texture improvements, thermal and non-thermal post-processing, and aesthetic improvements. Schneberger et al. [17] underscores the need for a comprehensive framework for the development of hybrid AM parts, which includes post-processing, testing, and life cycle monitoring. These studies collectively highlight the significance of post-processing in additive manufacturing for improving part quality and performance.

Focusing specifically on surface polishing, it's crucial to note that this process plays a significant role in enhancing the aesthetic and functional qualities of finished products, especially in additive manufacturing. Surface polishing helps in reducing surface roughness and improving the visual appeal of the product. It's not just about aesthetics; polished surfaces can also have improved wear resistance and may alter the product's interaction with its environment, like reducing friction or enhancing chemical resistance. This is particularly important in industries where precision and surface quality are paramount, such as in aerospace and medical devices.

Furthermore, surface polishing, particularly for products with complex shapes, presents a significant challenge in the manufacturing process. While CNC machines or robots are efficient at handling simpler shapes, the intricate nature and variability of complex geometries often necessitate human intervention. Automating these complex tasks is a challenge due to their intricacy. However, manual

polishing operations pose health risks and are physically demanding for human operators. Prolonged involvement in such tasks can lead to cognitive stress and physical strain, making continuous human engagement undesirable.

2.2. Framework for Post Processing

To address these challenges, the development of a robotic framework that can easily and efficiently perform these tasks on complex shapes is essential. This kind of framework should aim to reduce the time and monetary costs associated with intensive programming currently needed for robotic polishing of complex surfaces and, moreover, it would alleviate the health risks posed to human operators.

The proposed robotic framework [18] (shown in Figure 1) is designed to automate the polishing of complex geometries efficiently. Leveraging advanced technologies and innovative approaches, such as those demonstrated in the SYMPLEXITY [19] project, this framework is intended to foster safer and more productive collaboration between humans and robots.

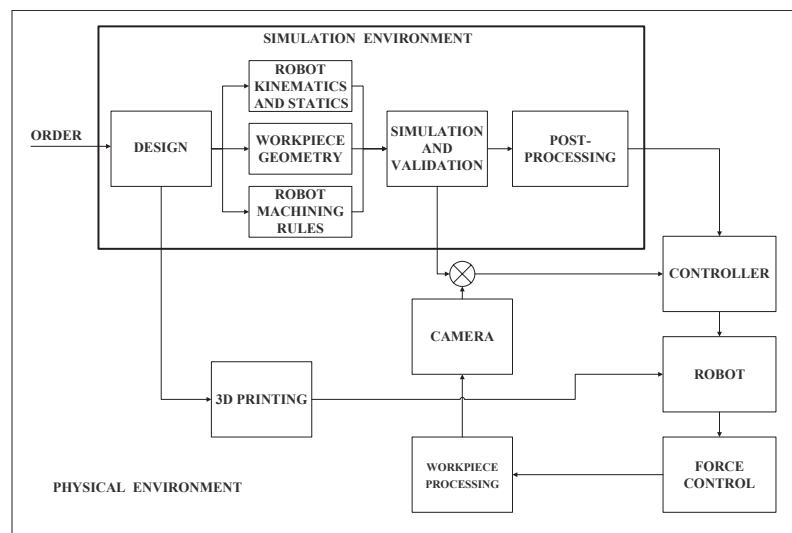


Figure 1. The illustration of the Framework for post-processing proposed in [18] shows the synergy between the simulation and physical environments. This framework highlights how virtual planning and real-world execution are integrated and interact in the field of robotic automation.

The Figure 1 illustrates the workflow of the proposed framework for the polishing task. The workflow begins with the Design phase (or CAD project phase) upon receiving an order. The component to be processed is inserted into the simulation environment (e.g., Siemens NX) and labeled as Workpiece Geometry phase. This is where the Robot Kinematics and the robot Statics are defined. Following this, the Robot Machining Rules are established, detailing how the robot should move and its constraints. Simulations are then conducted and validated. In the post-processing stage, the g-code for generating the Robot's base trajectory is created. This information is sent to the Controller, which manages the Robot. The actions of the robot are corrected in real-time based on feedback from a Camera sensors and Force Control, ensuring the finishing quality of the workpiece.

The primary objective of adopting this framework is to enhance the quality of the final product and ensure the well-being of human operators, avoiding having them perform repetitive and potentially dangerous tasks due to the production of dust and/or heat, while optimizing the overall efficiency of the manufacturing process.

3. System Overview

This section discusses key elements of the system, including the computers used, the philosophy and training process of the CNN network, and the use of GPUs to improve computing performance.

3.1. Images for Network Training

The resolution and size of the images are arbitrary. Standard camera sensors from different smartphones were used in environments with varying light conditions to acquire images for the training. During the training phase, the standard image sizes of the yolov3 network, specifically were used 416x416 and 608x608. In later stages, the training dimensions were modified. Furthermore, various parameters related to data augmentation were also changed in different training sets. More information can be found in the online repository of the project [20], where the various training sets have been divided and commented.

3.2. System Configuration and Preparation for Training

The initial training phase of the network took place on a workstation with higher computational capabilities compared to the final computer intended for operational use of the network. This approach optimize the training process, making it faster and more efficient.

In the preliminary phase, the network was created and tested on the less powerful computer, which is also the final device for use. The initial tests not only allowed for the customization and adaptation of the network to the specific performances of the final computer but also guided the creation of the network final structure. The result of this phase was the configuration of the network, ready for training. Figure 2 illustrates the general workflow adopted in this project, highlighting the key steps in the network development and training process.

Subsequently, this configured network was transferred to the more powerful workstation. In this phase, the primary focus was on training the weights of the network, leveraging the greater computational resources of the workstation for faster and more accurate training. This process ensured that the network, once trained, was optimized for performance on the less powerful computer.

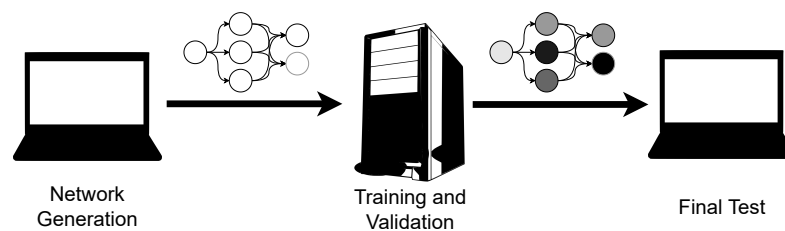


Figure 2. This image depicts the workflow where the network is initially created on a less powerful computer, then transferred to a high-performance workstation for weight training, and finally moved back to the original, less powerful computer for operational use. This process highlights the development strategy, focusing on creating a network that is adaptable and optimized for different computational environments.

3.3. System Performance

In the following Table 1, the system configurations used in the project are presented, highlighting the differences between the PC used for training and the one used for operational purposes.

Table 1. System Configurations for AI Vision Applications.

Designation	Operating System	Kernel Version	GPU Model	GPU Architecture	GPU Driver Version	CUDA Version	CUDA Driver Number
Training	Ubuntu 22.04.3 LTS	6.2.0-37-generic	NVIDIA GeForce RTX 3060 Lite Hash Rate	Ampere	470.223.02	12.2	3584
User	Ubuntu 16.04 xenial	4.15.0-142-generic	NVIDIA GeForce 710M + Intel HD Graphics	Fermi	384.130	-	96

Note: The configurations are specifically tailored for AI vision applications.

The system for training is equipped with a modern and powerful GPU, suitable for intensive tasks such as neural network training. In contrast, the PC for operational use incorporates an older GPU, based on NVIDIA Fermi architecture, which has computational performance limitations compared to more recent models.

Despite these limitations, tests have shown that satisfactory results can be achieved even on less advanced hardware by leveraging GPU processing. In particular, it was possible to maintain a sampling rate of around 15 FPS, demonstrating that effective performance for vision applications can be achieved even with older components. This aspect is particularly relevant for small and medium-sized enterprises, where investment in the latest hardware can be prohibitive, but there is still a desire to implement technologically advanced solutions.

3.3.1. CUDA and cuDNN

A crucial aspect for achieving efficient results in terms of FPS in detection on the user PC and reducing training times on the training PC is the optimization of computing performance. The use of CUDA drivers and cuDNN libraries was fundamental in enhancing performance. Comparative results between the use of parallel computing and non-parallel methods are shown in paragraph 6. Their application in improving the performance of framework Darknet for vision applications, particularly with YOLOv3, is significant.

CUDA (Compute Unified Device Architecture) is a parallel computing architecture developed by NVIDIA, which allows for a drastic increase in computing performance by leveraging the power of GPUs. This is particularly useful in vision applications like YOLOv3, which require intense processing for real-time object recognition and classification. Additionally, cuDNN (CUDA Deep Neural Network), a library of primitives for deep neural networks, further optimizes CUDA-based operations for deep learning. The integration of CUDA and cuDNN into Darknet, an open-source framework for neural networks, significantly accelerates the training and inference process of neural networks, also improving accuracy and efficiency in object detection with YOLOv3, making it a powerful tool for advanced vision applications.

There are several studies that highlight the importance of using GPU processors over CPUs with CUDA drivers and cuDNN libraries for executing AI algorithms. For example, Wang et al. [21] demonstrated that using GPUs to accelerate existing frameworks is an effective approach. They implemented a Single Shot Multibox Detector (SSD) with CUDA on GPUs, improving object detection speed by 22.5%. Singh et al. [22] developed a Deep Convolutional Neural Network (DCNN) on the CUDA platform for digit recognition, reducing the computation time of Artificial Neural Networks (ANNs) and increasing accuracy. They showed that the use of GPU hardware drastically reduces computation time, with a decrease of 96.51% for MNIST and 95.27% for EMNIST compared to using

a dual-core CPU. Pendlebury et al. [23] ran the "Mine Hunter" program, created by the authors, a parallel neural network simulator executed on NVIDIA CUDA. The simulation, performed on an Intel Quad Core i5-2500 3.3GHz system with 2 Nvidia GeForce GTX 480s, involved 128 "mine hunters" in a minefield with 8192 mines. The results show that the use of CUDA improves performance by up to 80% compared to an equivalent CPU implementation. These are just a few examples that demonstrate the importance of GPUs and CUDA drivers in achieving good speed performance.

3.4. Network Training Procedure

For the training of the custom Darknet Yolo network, a standard methodology was adopted, which includes the creation of a dataset, labeling, training, and subsequent model development. Figure 3 illustrates the logical sequence of operations carried out.

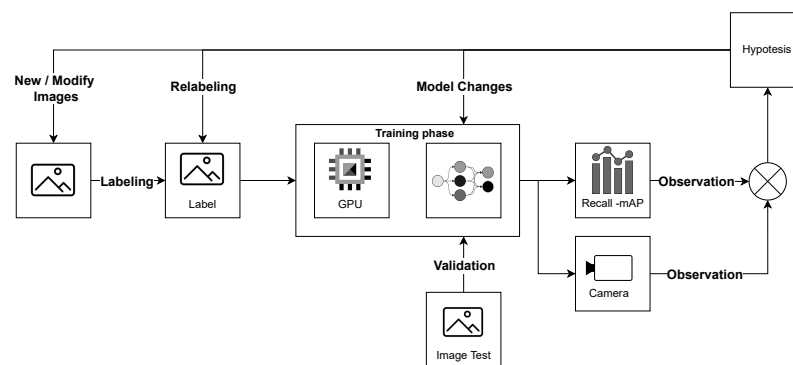


Figure 3. The figure represents the main phases of the neural network training process: dataset creation with associated labeling, training, model development, and validation. The validation phase emphasizes tests on real components (represented by a camera in the figure) and the evaluation of key parameters such as mean Average Precision (mAP) and Recall.

During the training process, each phase underwent targeted variations. Among the best practices adopted, initial overfitting on a limited number of images is highlighted, followed by subsequent network modeling through the addition of more photos [24]. This approach allowed for refining the accuracy and effectiveness of the model.

4. Dataset Description

In the field of artificial intelligence applied to vision, it is common to find datasets composed of thousands or even millions of images of various sizes. The vast availability of images on the internet facilitates the training of models that can be better "fitted," allowing for more effective generalization.

However, in the field of industrial manufacturing, the opposite problem is often encountered: a low need for generalization. In this sector, there is a tendency to work with more limited datasets, which do not require a strong capacity for generalization and allow the use of smaller models with a fewer number of images.

Recent studies have highlighted the potential of deep learning in applications with small datasets in the context of Industry 4.0. Chaoub et al. [25] emphasized the need for models that can perform well with limited data, particularly in the context of prognosis and health management of equipment. Şimşek et al. [26] further emphasized the role of deep learning in solving various industrial challenges, focusing on the ability of machines to communicate and make decisions. Kapusi et al. [27] provided a practical example of this, demonstrating the application of deep learning in real-time object detection for an industrial SCARA robot. These studies collectively highlight the potential of deep learning in applications with small datasets within the context of Industry 4.0.

In this study, only 26 images were used, which are available in the online repository of the project [20]. At the time of writing this article, the authors did not find an online a specialized dataset related to this type of defect detection. Therefore, this could be the first time that a datasets related to this

type of issue have been published. The following paragraph will show how the samples used for the generation of the dataset were obtained.

4.1. Procedure to Obtain Images for the Dataset

As there were no online photographs suitable for training on the specific defects of interest, we created this dataset from scratch, focusing on surface defects in aluminum materials. To produce an initial set of images, we first performed manual polishing of the aluminum samples to achieve a uniform surface finish. Subsequently, we restored unpolished areas by applying hydrochloric acid to accelerate the oxidation process. Figure 4 shows an example of the images collected, where the contrast between the polished and intentionally untreated areas is evident.



Figure 4. Sample of aluminum post-manual polishing. The image highlights the differences in surface finishing, with particular attention to the unpolished areas intentionally restored to detect typical material defects. This images was used for the network training, demonstrating the necessary contrast for accurate defect identification.

The dataset obtained from this process has been published in the online GitHub repository SPADD (Small-set Polished Aluminum Defect Detection) Dataset [20].

5. Network Design

This paragraph describes the custom Darknet YOLOv3 network and the construction of our configuration file used for training and detection.

5.1. Network Description

The design of the network aimed for high operational efficiency, even on older computers, as specified in the third paragraph 3. In these lines, we provide an overview of the CNN developed for the project. The description here is concise; for more in-depth information and technical details, please refer to the GitHub repository [20]. We will first illustrate the key components of the network, the convolutional and max pooling layers, and then explore the overall structure and logic of the network.

5.1.1. Convolutional Layers

The convolutional layers of the CNN apply a set of filters (or kernels) to the input to extract important features from images. Each filter in a convolutional layer is learned during training and becomes specialized in recognizing specific types of visual features, such as edges, corners, textures, or even more complex patterns depending on the depth of the network. The parameters used in this network are:

- **Size:** The size of the kernel (e.g., 3x3).
- **Filters:** The number of filters, which also determines the number of features extracted and the depth of the output volume.
- **Stride:** The number of pixels by which the filter moves at each step.

- **Padding:** Adds pixels to the edges of the input to allow the filter to work on the edges.
- **Activation:** The activation function, often ReLU or variants (like leaky ReLU), which introduces non-linearity.

5.1.2. Max Pooling Layers

Max pooling layers serve to reduce the spatial dimensionality (height and width) of the input while preserving important features. This reduces the number of parameters and computational load, contributing to the prevention of overfitting. The parameters used in this network are:

- **Size:** The size of the pooling window (e.g., 2x2).
- **Stride:** The number of pixels by which the pooling window moves at each step.

5.2. Network Functioning

The operation of the network is based on the interaction of the convolutional layers with each other and with the max pooling layers. As illustrated in **Figure 5**, the CNN used in this study is characterized by its essential and functional structure, which facilitates effective data processing while maintaining reduced complexity.

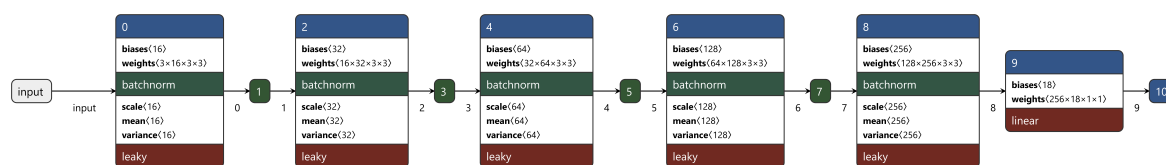


Figure 5. This illustration shows the architecture of the Convolutional Neural Network (CNN) of this project, using the online tool Netron [to be cited]. The blue layers numbered 0, 2, 4, 6, 8, and 9 represent the convolutional layers, each followed by a batch normalization layer, indicated by the upper red stripe, which serves to stabilize and accelerate learning. The green layers numbered 1, 3, 5, and 7 are the pooling layers, specifically max pooling, which reduce the spatial dimensionality of the data to decrease overfitting and improve computational efficiency. Finally, the blue layer number 10 indicates a fully connected layer, named the region layer, which performs the final classification or regression based on the features processed by the previous layers of the network.

The network begins with relatively small filters (3x3), a common practice for capturing fine details in images. As we progress through the network, the number of filters increases (from 16 to 256). Starting with a low number of filters and then gradually increasing is a standard procedure commonly used in neural networks: initially, the network looks for simple features (such as edges), and in later stages, identifies more complex characteristics using information abstracted from previous levels.

The use of the "leaky" activation function helps to avoid (or reduce) the problem of "dead" neurons that can occur with ReLU, improving the model's learning capacity [28,29].

Each max pool layer reduces the dimensions of the input by a factor of 2. This not only reduces the computational load and the number of parameters (preventing overfitting), but also allows filters in the subsequent convolutional layers to have a wider "receptive field." In other words, they can "see" and integrate information over larger areas of the image, allowing the network to recognize and combine features at different scales.

6. Results and Conclusion

Dataset Development and Network Training: This article describes the creation of a new dataset obtained from camera sensors for the recognition of superficial defects, utilizing open-source tools such as Darknet YOLOv3 [30] for network training and Yolo_mark [31] for image labeling. The process of network generation and training has been detailed, with all research outputs freely shared in the developing online repository, SPADD [20].

Dataset Evaluation: Considering the small size of the network used in this paper, an evaluation using standard methodologies such as mAP and Recall is, in the opinion of the authors, not reliable because the results can be easily masked through targeted training. Instead, it was preferred to demonstrate as the result of this analysis a test video that shows how the network is able to detect defects in a test video, following a subsequent limited targeted training, starting from the weights present in the Git repository. The videos are also available in the Git repository [20].

Network Performance in Constrained Environments: Our customized network has demonstrated effective functionality in computationally constrained environments, as per the 'user PC' specifications. A detection frame rate of 15 FPS was achieved with a GPU, processing images of size 416x416, under test conditions using only the Darknet terminal application without any background processes.

Dataset Limitations and Call for Contributions: The current dataset has limitations, particularly in its capacity for comprehensive generalization assessment. To enhance the dataset, we encourage readers to contribute relevant images to the SPADD project's official repository [20].

Training Methodology and Data Augmentation: The network's training employed "data augmentation" techniques, modifying saturation, hue, blur, exposure, mosaic, and randomizing image dimensions. This approach aims to enhance the generalizability of network features despite the constraints of a limited dataset and network size. Training was segmented into multiple sets, each ending with modifications to the dataset or configuration file, based on the outcomes of each training phase.

Application and Importance in Industrial Contexts: This methodology was evaluated for its feasibility in SMEs, which typically prioritize low initial and maintenance costs over high automation and production speed. The study delves into how technologies like cobots and AI can be effectively and rapidly deployed in smaller production settings. Additionally, it highlights the challenges of recognizing light reflection defects, influenced by factors like warehouse exposure and weather conditions, necessitating continuous training or structured environments. Investigating the feasibility, advantages, and drawbacks of employing AI for surface defect recognition in industrial settings is crucial in the current manufacturing landscape.

Future Integration with Collaborative Robot Polishing Tasks: Building upon the results of this study, one of the forthcoming steps is to extend the application of our defect detection system to direct collaborative robots (cobots) in executing polishing tasks. This advancement is aimed at demonstrating the practicality and feasibility of such an integrated system for small and medium-sized enterprises (SMEs). By implementing sensing results in cobot-assisted polishing operations, it is intended to provide SMEs with an efficient and cost-effective solution that improves the quality of their production processes. This advancement represents a step towards the practical application of AI and robotics in industry, particularly in optimizing tasks that have traditionally been challenging to automate.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "Conceptualization, A.B.; methodology, A.B.; software, A.B.; validation, A.B.; formal analysis, A.B.; investigation, A.B.; resources, A.B.; data curation, A.B.; writing—original draft preparation, A.B.; writing—review and editing, M.C.P.; visualization, A.B.; supervision, M.C.P.; project administration, M.C.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: <https://github.com/AlbinEV/SPADD-Dataset>.

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

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
CNN	Convolutional Neural Network
FPS	Frames Per Second

GPU	Graphics Processing Unit
CUDA	Compute Unified Device Architecture
cuDNN	CUDA Deep Neural Network
SSD	Single Shot Multibox Detector
DCNN	Deep Convolutional Neural Network
ANNs	Artificial Neural Networks
MNIST	Modified National Institute of Standards and Technology database
EMNIST	Extended MNIST
RAS	Robotic Autonomous Systems
SM	Smart Manufacturing
ICTs	Information and Communication Technologies
SMEs	Small and Medium-Sized Enterprises

References

1. Garcia, M.; Rauch, E.; Salvalai, D.; Matt, D. AI-based human-robot cooperation for flexible multi-variant manufacturing, 2021.
2. Sowa, K.; Przegalinska, A.; Ciechanowski, L. Cobots in knowledge work: Human - AI collaboration in managerial professions. *Journal of Business Research* **2021**, *125*, 135–142. <https://doi.org/10.1016/j.jbusres.2020.11.038>.
3. Bohušík, M.; Stenclák, V.; Císar, M.; Bulej, V.; Kuric, I.; Dodok, T.; Bencel, A. Mechatronic Device Control by Artificial Intelligence, 2023. <https://doi.org/10.3390/s23135872>.
4. Bhardwaj, A.; Kishore, S.; Pandey, D.K. Artificial Intelligence in Biological Sciences, 2022. <https://doi.org/10.3390/life12091430>.
5. Nanjangud, A.; Blacker, P.C.; Bandyopadhyay, S.; Gao, Y. Robotics and AI-Enabled On-Orbit Operations With Future Generation of Small Satellites. *Proceedings of the IEEE* **2018**, *106*, 429–439. <https://doi.org/10.1109/JPROC.2018.2794829>.
6. Carbonari, L.; Forlini, M.; Scoccia, C.; Costa, D.; Palpacelli, M.C. Disseminating Collaborative Robotics and Artificial Intelligence Through a Board Game Demo. In Proceedings of the 2022 18th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA). IEEE, nov 2022. <https://doi.org/10.1109/mesa55290.2022.10004477>.
7. Kaczmarek, W.; Lotys, B.; Borys, S.; Laskowski, D.; Lubkowski, P. Controlling an Industrial Robot Using a Graphic Tablet in Offline and Online Mode, 2021. <https://doi.org/10.3390/s21072439>.
8. Pandiyan, V.; Shevchik, S.; Wasmer, K.; Castagne, S.; Tjahjowidodo, T. Modelling and monitoring of abrasive finishing processes using artificial intelligence techniques: A review. *Journal of Manufacturing Processes* **2020**, *57*, 114–135. <https://doi.org/10.1016/j.jmapro.2020.06.013>.
9. Kim, D.H.; Kim, T.J.Y.; Wang, X.; Kim, M.; Quan, Y.J.; Oh, J.W.; Min, S.H.; Kim, H.; Bhandari, B.; Yang, I.; et al. Smart Machining Process Using Machine Learning: A Review and Perspective on Machining Industry. *International Journal of Precision Engineering and Manufacturing-Green Technology* **2018**, *5*, 555–568. <https://doi.org/10.1007/s40684-018-0057-y>.
10. Davis, J.; Edgar, T.; Graybill, R.; Korambath, P.; Schott, B.; Swink, D.; Wang, J.; Wetzel, J. Smart Manufacturing. *Annu. Rev. Chem. Biomol. Eng.* **2015**, *6*, 141–160. <https://doi.org/10.1146/annurev-chembioeng-061114-123255>.
11. Mittal, S.; Khan, M.A.; Romero, D.; Wuest, T. Smart manufacturing: Characteristics, technologies and enabling factors. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* **2017**, *233*, 1342–1361. <https://doi.org/10.1177/0954405417736547>.
12. Luo, Q.; Fang, X.; Liu, L.; Yang, C.; Sun, Y. Automated Visual Defect Detection for Flat Steel Surface: A Survey. *IEEE Transactions on Instrumentation and Measurement* **2020**, *69*, 626–644. <https://doi.org/10.1109/TIM.2019.2963555>.
13. Bajrami, A.; Palpacelli, M.C. A Proposal for a Simplified Systematic Procedure for the Selection of Electric Motors for Land Vehicles with an Emphasis on Fuel Economy. *Machines* **2023**, *11*, 420. <https://doi.org/10.3390/machines11040420>.
14. Peng, X.; Kong, L.; Fuh, J.Y.; Wang, H. A Review of Post-Processing Technologies in Additive Manufacturing, 2021. <https://doi.org/10.3390/jmmp5020038>.

15. Lane, B.; Moylan, S.; Whittenton, E. Post-process machining of additive manufactured stainless steel. Proceedings of the 2015 ASPE Spring Topical Meeting: Achieving Precision Tolerances in Additive Manufacturing, Raleigh, NC, 2015, [Proceedings of the 2015 ASPE Spring Topical Meeting: Achieving Precision Tolerances in Additive Manufacturing, Raleigh, NC].
16. Mishra, P.; Sood, S.; Pandit, M.; Khanna, P. Additive Manufacturing: Post Processing Methods and Challenges. *Advanced Engineering Forum* **2021**, *39*, 21–42. <https://doi.org/10.4028/www.scientific.net/AEF.39.21>.
17. Schneberger, J.H.; Kaspar, J.; Vielhaber, M. Post-processing and testing-oriented design for additive manufacturing - A general framework for the development of hybrid AM parts. *Procedia CIRP* **2020**, *90*, 91–96. <https://doi.org/10.1016/j.procir.2020.01.059>.
18. Bajrami, A.; Palpacelli, M.C. A Flexible Framework for Robotic Post-Processing of 3D Printed Components. In Proceedings of the Volume 7: 19th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA). American Society of Mechanical Engineers, 2023, IDETC-CIE2023. <https://doi.org/10.1115/detc2023-109746>.
19. Symbiotic Human-Robot Solutions for Complex Surface Finishing Operations. <https://cordis.europa.eu/project/id/637080>. Accessed: 2023-12-13.
20. SPADD-Dataset: Early set of varied images displaying unpolished surface defects on aluminum. <https://github.com/AlbinEV/SPADD-Dataset>. Accessed: 2023-12-13.
21. Wang, C.; Endo, T.; Hirofuchi, T.; Ikegami, T. Speed-up Single Shot Detector on GPU with CUDA. In Proceedings of the 2022 23rd ACIS International Summer Virtual Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD-Summer). IEEE, jul 2022. <https://doi.org/10.1109/snspd-summer57817.2022.00015>.
22. Singh, S.; Paul, A.; Arun, M. Parallelization of digit recognition system using Deep Convolutional Neural Network on CUDA. In Proceedings of the 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS). IEEE, may 2017. <https://doi.org/10.1109/ssps.2017.8071623>.
23. Pendlebury, J.; Xiong, H.; Walshe, R. Artificial Neural Network Simulation on CUDA. In Proceedings of the 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications. IEEE, oct 2012. <https://doi.org/10.1109/ds-rt.2012.40>.
24. A Recipe for Training Neural Networks. <https://karpathy.github.io/2019/04/25/recipe/>. Accessed: 2023-12-19.
25. Chaoub, A.; Cerisara, C.; Voisin, A.; Iung, B. Deep Learning Representation Pre-training for Industry 4.0. *PHM Society European Conference* **2022**, *7*, 571–573. <https://doi.org/10.36001/phme.2022.v7i1.2784>.
26. Şimşek, M.A.; Orman, Z., A Study on Deep Learning Methods in the Concept of Digital Industry 4.0. In *Advances in E-Business Research*; IGI Global, 2021; pp. 318–339. <https://doi.org/10.4018/978-1-7998-5015-1.ch016>.
27. Kapusi, T.P.; Erdei, T.I.; Husi, G.; Hajdu, A. Application of Deep Learning in the Deployment of an Industrial SCARA Machine for Real-Time Object Detection. *Robotics* **2022**, *11*, 69. <https://doi.org/10.3390/robotics11040069>.
28. Jiang, T.; Cheng, J. Target Recognition Based on CNN with LeakyReLU and PReLU Activation Functions. In Proceedings of the 2019 International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC). IEEE, aug 2019. <https://doi.org/10.1109/sdpc.2019.00136>.
29. Mastromichalakis, S. ALReLU: A different approach on Leaky ReLU activation function to improve Neural Networks Performance, 2020. <https://doi.org/10.48550/ARXIV.2012.07564>.
30. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**.
31. Github, AlexeyAB/Yolo_mark. https://github.com/AlexeyAB/Yolo_mark. Accessed: 2023-12-19.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.