machines

MDPI

Article

From Dataset Creation to Defect Detection: A Proposed Procedure for a Custom CNN Approach for Polishing Applications on Low-Performance PCs

Albin Bajrami *,† **and Matteo Claudio Palpacelli**

Department of Industrial Engineering and Mathematical Sciences, Università Politecnica delle Marche, 60131 Ancona, Italy; m.c.palpacelli@staff.univpm.it

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

† The Italian Doctorate in Robotics and Intelligent Machines.

Abstract: This study focuses on training a custom, small Convolutional Neural Network (CNN) using a limited dataset through data augmentation that is aimed at developing weights for subsequent fine-tuning on specific defects, namely improperly polished aluminum surfaces. The objective is to adapt the network for use in computationally restricted environments. The methodology involves using two computers—a low-performance PC for network creation and initial testing and a more powerful PC for network training using the Darknet framework—after which the network is transferred back to the initial low-performance PC. The results demonstrate that the custom lightweight network suited for a low-performance PC effectively performs object detection under the described conditions. These findings suggest that using tailored lightweight networks for recognizing specific types of defects is feasible and warrants further investigation to enhance the industrial defect detection processes in limited computational settings. This approach highlights the potential for deploying AI-driven quality control in environments with constrained hardware capabilities.

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

Citation: Bajrami, A.; Palpacelli, M.C. From Dataset Creation to Defect Detection: A Proposed Procedure for a Custom CNN Approach for Polishing Applications on Low-Performance PCs. *Machines* **2024**, *12*, 453. https://doi.org/10.3390/machines12070453

Academic Editors: Davide Astolfi and Ahmed Abu-Siada

Received: 30 April 2024
Revised: 29 May 2024
Accepted: 28 June 2024
Published: 2 July 2024

1. Introduction

Technological progress in the fields of Artificial Intelligence (AI) and Robotics is radically changing the industrial landscape, thus 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.

This article examines the challenges and methodologies associated with training neural networks on small datasets for industrial applications, with the aim of efficiently recognizing and classifying defects on reflective surfaces. The focus is on developing a streamlined process that allows smaller and customized networks to be deployed effectively in industrial environments, where computational resources and data availability are often limited.

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. The incorporation of advanced sensors and AI algorithms into today's cobots enhances

their adaptability across diverse tasks. For instance, the integration of AI in cobots, as discussed by Garcia et al. [1], optimises flexible manufacturing, thus allowing robots to adjust dynamically to production changes without extensive manual reprogramming. This not only boosts efficiency but also fosters innovation in industrial automation.

The influence of AI extends across various fields, thus significantly altering problem-solving methods. Sowa et al. [2] emphasise the advantages of human–AI collaboration in management, thus citing the positive response from employees and their concerns regarding full automation. Bohušík et al. [3] and Bhardwaj et al. [4] demonstrate the broad impact of AI in the fields of mechatronics and life sciences, respectively. Bohušík et al. utilised AI for precise motor control in kinematic mechanisms, thus enhancing performance by correcting mechanical inaccuracies. Bhardwaj et al. investigated the transformative role of AI in medical diagnostics, pharmaceuticals, precision agriculture, and bioindustrial processes. They emphasise the potential of AI to improve quality of life and reduce costs. Nanjangud et al. [5] highlight the pivotal role of AI in space exploration, from satellite servicing to debris removal. This is facilitated by advanced robotic systems that autonomously perform high-risk tasks. Carbonari et al. [6] present an example of AI in a playful context, thus utilising a robot for the game Connect Four. These are only some example that demonstrates the capabilities of AI in visual perception and decision making.

AI has brought about a revolution in manufacturing, even in postprocessing operations such as polishing and machining. Neural networks, which are networks of interconnected neurons, are being used to enable precise object recognition and classification. This enables the detection and reworking, if possible, of objects or surface defects. This has resulted in a more efficient manufacturing process, with a concomitant reduction in the production of defective artefacts. Furthermore, the simplification of programming has become a crucial aspect in the field of industrial robotics, thus facilitating the training and effective deployment of these systems. For instance, Kaczmarek et al. [7] developed an application that simplifies the programming of industrial robots by employing task-oriented programming, rather than step-by-step instructions. This facilitates faster and easier integration, which is a crucial factor in small- and medium-sized enterprises (SMEs). The utilisation of CNNs can streamline the training process of robots for decision making in situations where the desired outcome is not achieved.

The collective findings of Pandiyan et al. [8], Kim et al. [9], and Luo et al. [10] demonstrate the transformative role of AI in manufacturing. Pandiyan et al. provide a detailed account of the enhancements that AI has brought to abrasive finishing processes, thus resulting in improved accuracy, efficiency, and monitoring. This underscores the significance of smart manufacturing (SM) and optimized process management. Kim et al. explore the potential of machine learning in SM, thus noting its ability to increase sustainability and reduce environmental impacts through networked data and ICTs [11,12]. Luo et al. present a review of technologies for surface defect recognition in flat steel products, with a particular focus on quality control.

The short reviews presented here illustrate the growing influence of AI in a number of industries that require sophisticated decision making. However, there is a gap in the scientific literature on the topic of incorrectly polished surface recognition using CNN in industrial environments with small datasets and limited computing power. The use of small datasets is an increasingly central issue in AI deployment. To address this, few-shot learning (FSL) emerges as a promising approach, thus enabling effective learning from limited data [13–16]. In general, few-shot learning (FSL) involves the use of specialised architectures that have been designed to enable effective learning from minimal data. However, in this case, the objective is to train a 'classical' network, such as YOLOv3, with a small dataset and data augmentation to create a lightweight model for a transfer learning approach. In this manner, subsequent fine-tuning enables the model to be adapted to other small datasets, thereby aligning with the principles of few-shot learning.

This research addresses several issues related to the recognition of object features on their reflective surfaces through the use of neural networks, with a particular focus on

environments with limited computational capacity. Similar work can be found in [17], where the authors present WearNet, a lightweight NN for scratch detection. Other similar work can be found in [18,19], where a lightweight approach has been used in CNN but in other research areas, such as surveillance and medical applications.

The objective is to develop an efficient and viable approach to improve surface finishing processes in the industry, thus utilising the power of AI in a more accessible and cost-effective computational context. The main problems addressed in this research are the following:

- *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*: The creation of an evolving public dataset for the recognition of surface defects related to light reflection;
- *Defect Detection in Low-Computational-Capacity Environments*: This demonstration will illustrate how defect detection can be effectively performed on less powerful hardware, with a focus on the achievable performance metrics;
- *Network Development Methodology*: A description of the adopted methodology, including the creation of the network on lower performing PCs and the training of weights on more powerful devices;
- *Creation and Publication of Weights for Transfer Learning*: The development and sharing of weights to facilitate transfer learning;

The paper is structured as follows: Section 2 discusses the challenges in polishing tasks and the necessity of a robotic framework for automating postproduction processes. Section 3 provides an overview of the system detailing components, configurations, and performance analysis, including hardware optimizations like CUDA and cuDNN. Section 4 describes the dataset creation and the procedure of the dataset creation. Section 5 explores the design of the custom network, thus detailing the roles of various layers and network architecture. Finally, Section 6 summarizes the study's results and conclusions, thus discussing the training process, data augmentation techniques, and the network's performance in limited computational settings. It concludes with future implications for collaborative robotic tasks in industrial contexts.

2. Flexible Framework for Polishing

In the first paragraph, we discussed postprocessing and highlighted the use of cobots and AI in this context. This section aims to define more specifically the aspects of postprocessing addressed in the article, and it presents the robotics framework designed to solve the problem of how to easily polish complex surfaces using cobots.

2.1. Challenges in Polishing Tasks

In manufacturing, postprocessing is a crucial stage that aims to enhance the quality of the final product by removing any undesired properties that may have been introduced during production. Surface polishing, a key postprocessing step, has the potential to significantly improve both the aesthetic and functional qualities of finished products [20–22]. Schneberger et al. [23] underscore the need for a comprehensive framework for the development of hybrid AM parts, which includes postprocessing, testing, and life cycle monitoring. However, polishing complex shapes presents challenges, thus often requiring human intervention due to the limitations of CNC machines and robots in handling intricate geometries.

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

that are currently needed for robotic polishing of complex surfaces and, moreover, it would alleviate the health risks posed to human operators.

The proposed robotic framework [24] (showen 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 [25] project, this framework is intended to foster safer and more productive collaboration between humans and robots.

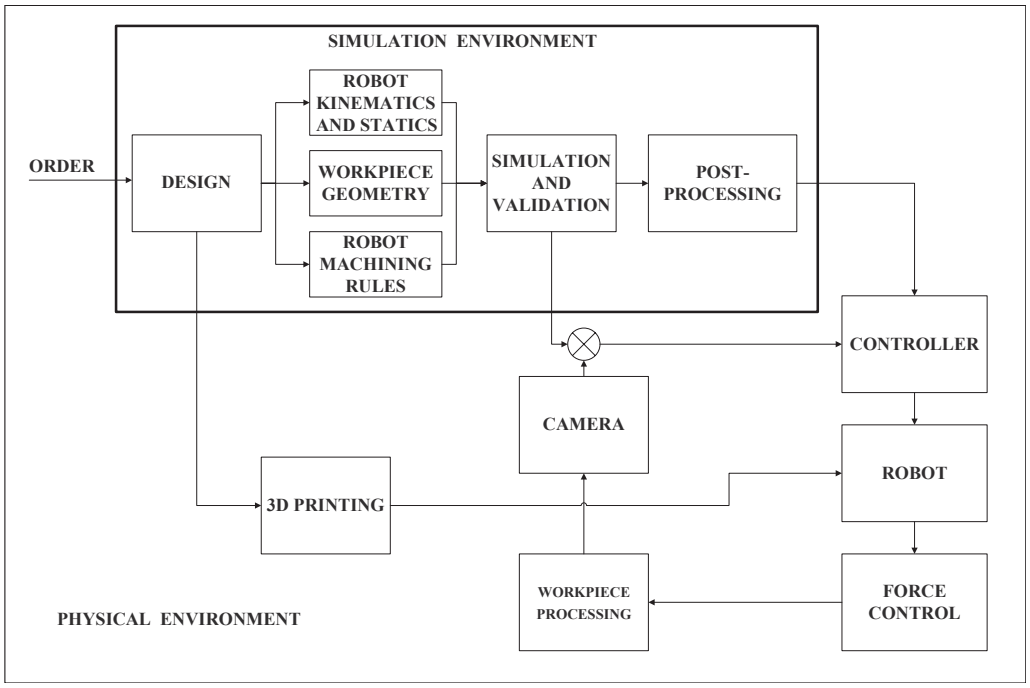

Figure 1. The illustration of the framework for postprocessing proposed in [24] shows the synergy between the simulation and physical environments. This framework highlights how virtual planning and real-world execution are integrated and interact in the field of robotic automation.

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, thus detailing how the robot should move and its constraints. Simulations are then conducted and validated. In the postprocessing 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 Camera sensors and Force Control, thus 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, thus avoiding having them perform repetitive and potentially dangerous tasks due to the production of dust and/or heat while optimising the overall efficiency of the manufacturing process.

3. System Overview

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

3.1. 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 was done to optimise the training process, thus making it faster and more efficient.

In the preliminary phase, the network was created and tested on the less powerful computer, which was also the final device for use. The initial tests not only allowed for the customisation 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, thus ready for training. Figure 2 illustrates the general workflow adopted in this project, thereby 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, as well as leveraging the greater computational resources of the workstation for faster and more accurate training. This process ensured that the network, once trained, was optimised for performance on the less powerful computer.

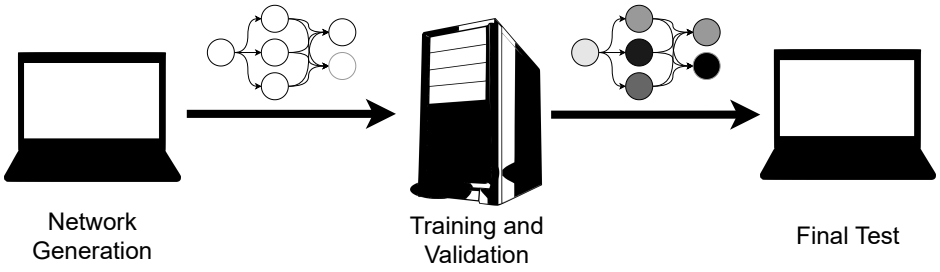

Figure 2. This image depicts the workflow, where the network is initially created on a less powerful computer (represented by white circles), then transferred to a high-performance workstation for weight training (gray-scale circles), and finally moved back to the original, less powerful computer for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

3.2. System Performance

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

Table 1. System configurations for AI vision spplications.

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

Despite these limitations, the 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, thus 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.

CUDA and cuDNN

The implementation of optimised computing performance enabled the achievement of efficient FPS in detection on the user PC and reduced training times on the training PC. The utilisation of CUDA drivers and cuDNN libraries was instrumental in enhancing the performance. Comparative results of parallel versus nonparallel computing are presented in Section 6, which demonstrate the significant performance boost in the vision applications with Darknet. CUDA, developed by NVIDIA, markedly enhances computing performance by capitalising on GPUs, which is of paramount importance for real-time object recognition in applications such as YOLOv3. cuDNN further optimises deep learning operations. Their integration into Darknet expedites training and inference, thus enhancing the accuracy and efficiency in object detection. A number of studies, including those by Wang et al. [26], Singh et al. [27], and Pendlebury et al. [28], have demonstrated the advantages of using GPUs with CUDA for AI algorithms. These studies have shown that GPUs offer significant improvements in speed and performance compared to CPUs.

3.3. Network Training Procedure

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

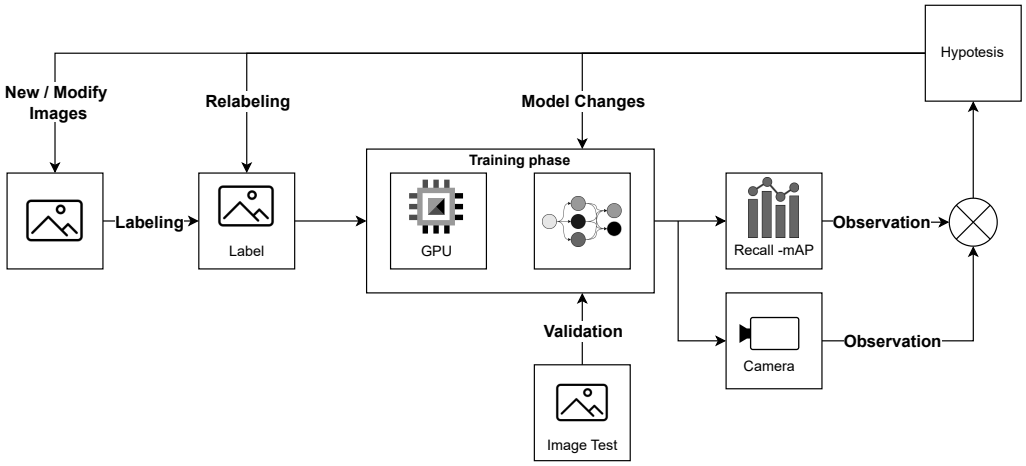

Figure 3. The figure represents the main phases of the neural network training process: dataset creation with associated labeling, training, model development, and validation. The validation phase emphasizes tests on real components (represented by a camera in the figure) and the evaluation of key parameters 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 was highlighted, followed by subsequent network modeling through the addition of more photos [29]. 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", thus allowing for more effective generalisation.

However, in the case of industrial manufacturing, the opposite situation is often encountered: a low need for generalisation. In this case, there is a tendency to work with

more limited datasets, which do not require a strong capacity for generalisation and allow for 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. Chaoub et al. [30] emphasised the need for models that can perform well with limited data, particularly in the context of prognosis and the health management of equipment. Şimşek et al. [31] further emphasised the role of deep learning in solving various industrial challenges, thus focusing on the ability of machines to communicate and make decisions. Kapusi et al. [32] provided a practical example of this, thus 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.

In this work, only 26 images were used, and some of the images that compose this dataset are shown in Figure 4. At the time of writing this article, the authors did not find an online specialized dataset related to this type of defect detection. Therefore, this could be the first time that a dataset related to this type of issue has been published. The following paragraph will show how the samples used for the generation of the dataset were obtained.

Figure 4. Sample of aluminum postmanual polishing used in the training phase. The image highlights the differences in surface finishing, with particular attention to the unpolished areas intentionally restored to detect typical material defects.

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, thus 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.

4.2. Dataset Images Description

The dataset comprises a substantial number of images captured using smartphones. The images were taken at a resolution of 1920 × 4160 pixels under controlled laboratory conditions with variable lighting to simulate both natural and artificial environments. The camera settings were generally standardised to an aperture of f/2.2 and a focal length of 4 mm. However, the ISO settings and shutter speeds were varied to adapt to different lighting conditions. The primary focus of this dataset is on unpolished areas on aluminium surfaces, which have been classified as the sole type of defect. The defects in question exhibit variation in terms of their appearance, size, and severity. These factors are of critical importance in ensuring that the neural network is able to identify and categorise polishing

defects with the requisite degree of accuracy. All images were manually annotated to identify and mark the unpolished areas, thereby ensuring the highest possible precision in the training set for our neural network. The manual annotation process was conducted using trained technicians, with the objective of ensuring accuracy and consistency across the entire dataset. In order to enhance the robustness of our model, several augmentation techniques were employed, including maintaining a fixed angle, adjusting saturation, setting exposure, and modifying hue. Additionally, the application of mosaic and flip effects was utilised. We also integrated mixup techniques with a specific setting for max chart loss during the initial training phase. It is crucial to highlight that no postprocessing was applied to the images following their initial capture and prior to being fed into the training pipeline. This approach was selected in order to maintain the authenticity of the defect characteristics and to challenge the network's ability to learn from raw data. The variability introduced through different lighting conditions and the comprehensive range of image augmentations ensure that our network has been trained on a dataset that closely mimics real-world scenarios. This methodology ensures the generality and reliability of our network in practical applications.

5. Network Design

This paragraph describes the custom Darknet 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, has been specified in the Section 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 [33]. 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 specialised 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 the following:

- Size: The size of the kernel (e.g., 3×3).
- 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 nonlinearity.

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, thus contributing to the prevention of overfitting. The parameters used in this network are the following:

- Size: The size of the pooling window (e.g., 2×2).
- 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.

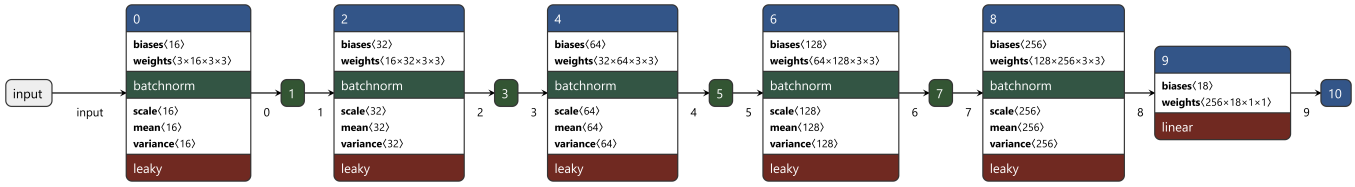

Figure 5. This illustration shows the architecture of the Convolutional Neural Network (CNN) of this project using the online tool Netron. The blue layers numbered 0, 2, 4, 6, 8, and 9 represent the convolutional layers, each followed by a batch normalization layer, indicated by the upper red stripe, which serves to stabilise and accelerate learning. The green layers numbered 1, 3, 5, and 7 are max pooling layers, 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 (3×3), which is 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, it 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, thus improving the model's learning capacity [34,35]. 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 it 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, thus allowing the network to recognize and combine features at different scales.

6. Results and Conclusions

The principal findings of this study are presented below. It should be noted that the continuous references to links in the online repository are not essential for comprehension of the results and are intended to facilitate the sharing of all the results in an open and transparent manner.

Dataset Development and Network Training: This article describes the creation of a new dataset obtained from camera sensors for the recognition of superficial defects, thus utilizing open source tools such as Darknet [36] for network training and Yolo_mark [37] for image labelling. The process of network generation and training has been detailed, with all research outputs freely shared in the developing online repository SPADD [33].

Dataset Evaluation: Considering the small size of the network used in this paper, an evaluation using standard methodologies such as mAP and Recall may not be entirely reliable, as the results can be easily influenced through targeted training. However, we present the values obtained using the standard parameters typically employed in training. We conducted a comparative analysis between our custom lightweight CNN and yolov3-tiny, which is the network from which our model is derived. The results of these trainings are summarised in Table 2. While standard metrics are reported, we believe that, given the small dataset and network size, these metrics may not fully represent the network's real-world performance.

While the results provide some insight, they should be interpreted with caution due to the limitations of the dataset size and the scale of the network. The videos demonstrating

the network's defect detection capabilities are available in the Git repository [33]. For a visual representation of the model's predictions, see Figure 6.

Table 2. Comparative results of training our custom lightweight CNN with and without data augmentation.

Metric	With Data Augmentation	Without Data Augmentation
mAP	76.37%	95.02%
Recall	0.6923	0.9500
Precision	0.9730	0.9560
F1 Score	0.81	0.95
Average IoU	76.17%	73.62%

Note: Performance metrics shown here have been obtained under standard training parameters, yet due to constraints such as the small dataset and network size, these figures might not fully encapsulate real-world efficacy.

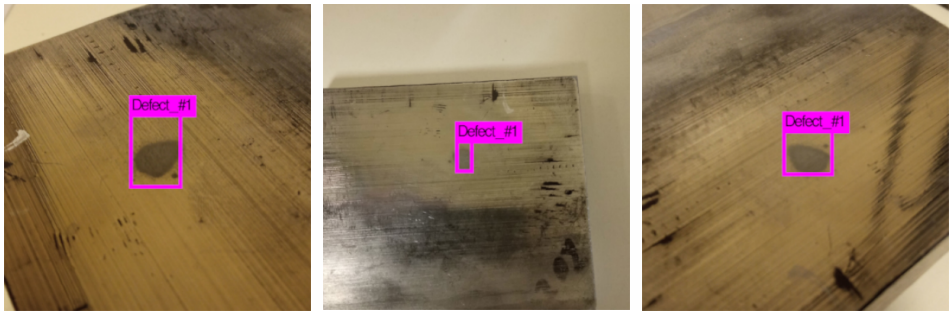

Figure 6. Example of model predictions on test data.

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 416 × 416, under test conditions using only the Darknet terminal application without any background processes. The Table 3 below compares the use of CPUs and GPUs, thus emphasizing the benefits of implementing GPU-based vision.

Table 3. Comparison of network speeds using CPU and GPU.

Network Size (Px)	Speed with CPU (FPS)	Speed with GPU (FPS)
32 × 32	15	15
160 × 160	12	15
320 × 320	3	15
416 × 416	2	15
608 × 608	<1	8

Note: Speed measurements are in frames per second (FPS).

Dataset Limitations and Network Characteristics: The current dataset has limitations, particularly in its capacity to provide a comprehensive assessment of generalisation across various conditions. The performance was highly dependent on the hardware used and was further constrained by the small size of the dataset, which limits its ability to generalise well when faced with new types of defects, such as oxidation on different materials. Despite these challenges, it is important to note that our network, while yielding lower performance metrics compared to more robust models like YOLOv3-tiny, benefits significantly from its substantially lower computational load. This makes it a viable option for applications where computational resources are limited. To enhance the dataset and improve the network's generalisation capabilities, we encourage readers to contribute relevant images to the SPADD project's official repository [33].

Training Methodology and Data Augmentation: The network's training employed "data augmentation" techniques, thus modifying the angle, saturation, exposure, hue, blur, flip, mixup, and mosaic, as well as randomizing the image dimensions. This approach aims

to enhance the generalisability of the network features despite the constraints of a limited dataset and network size. The 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 prioritise 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 recognising light reflection defects, which are influenced by factors like warehouse exposure and weather conditions, thus 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 toward the practical application of AI and robotics in industry, particularly in optimizing tasks that have traditionally been challenging to automate.

Author Contributions: Conceptualisation, 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.; visualisation, 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 (accessed on 1 July 2024).

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

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of Open Access ournals
TLA	Three-Letter Acronym
LD	Linear Dichroism
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. In Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management, Singapore, Singapore, 7–11 March 2021.
2. Sowa, K.; Przegalinska, A.; Ciechanowski, L. Cobots in knowledge work: Human—AI collaboration in managerial professions. *J. Bus. Res.* **2021**, *125*, 135–142. [CrossRef]
3. Bohušík, M.; Stenchlák, V.; Císar, M.; Bulej, V.; Kuric, I.; Dodok, T.; Bencel, A. Mechatronic Device Control by Artificial Intelligence. *Sensors* **2023**, *23*, 5872. [CrossRef] [PubMed]
4. Bhardwaj, A.; Kishore, S.; Pandey, D.K. Artificial Intelligence in Biological Sciences. *Life* **2022**, *12*, 1430. [CrossRef] [PubMed]
5. Nanjangud, A.; Blacker, P.C.; Bandyopadhyay, S.; Gao, Y. Robotics and AI-Enabled On-Orbit Operations With Future Generation of Small Satellites. *Proc. IEEE* **2018**, *106*, 429–439. [CrossRef]
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), Taipei, Taiwan, 28–30 November 2022 ; IEEE: Piscataway, NJ, USA, 2022. [CrossRef]
7. Kaczmarek, W.; Lotys, B.; Borys, S.; Laskowski, D.; Lubkowski, P. Controlling an Industrial Robot Using a Graphic Tablet in Offline and Online Mode. *Sensors* **2021**, *21*, 2439. [CrossRef] [PubMed]
8. Pandiyan, V.; Shevchik, S.; Wasmer, K.; Castagne, S.; Tjahjowidodo, T. Modelling and monitoring of abrasive finishing processes using artificial intelligence techniques: A review. *J. Manuf. Process.* **2020**, *57*, 114–135. [CrossRef]
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. *Int. J. Precis. Eng. Manuf.-Green Technol.* **2018**, *5*, 555–568. [CrossRef]
10. Luo, Q.; Fang, X.; Liu, L.; Yang, C.; Sun, Y. Automated Visual Defect Detection for Flat Steel Surface: A Survey. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 626–644. [CrossRef]
11. 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. [CrossRef]
12. Mittal, S.; Khan, M.A.; Romero, D.; Wuest, T. Smart manufacturing: Characteristics, technologies and enabling factors. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2017**, *233*, 1342–1361. [CrossRef]
13. Rezaei, M.; Diepeveen, D.; Laga, H.; Jones, M.G.; Sohel, F. Plant disease recognition in a low data scenario using few-shot learning. *Comput. Electron. Agric.* **2024**, *219*, 108812. [CrossRef]
14. Duan, R.; Li, D.; Tong, Q.; Yang, T.; Liu, X.; Liu, X. A survey of few-shot learning: An effective method for intrusion detection. *Secur. Commun. Netw.* **2021**, *2021*, 4259629. [CrossRef]
15. Parnami, A.; Lee, M. Learning from few examples: A summary of approaches to few-shot learning. *arXiv* **2022**, arXiv:2203.04291.
16. Chang, M.C.; Alaeddini, A. Few-shot classification with prototypical neural network for hospital flow recognition under uncertainty. *Netw. Model. Anal. Health Inform. Bioinform.* **2024**, *13*, 19. [CrossRef]
17. Li, W.; Zhang, L.; Wu, C.; Cui, Z.; Niu, C. A new lightweight deep neural network for surface scratch detection. *Int. J. Adv. Manuf. Technol.* **2022**, *123*, 1999–2015. [CrossRef]
18. Ullah, A.; Muhammad, K.; Ding, W.; Palade, V.; Haq, I.U.; Baik, S.W. Efficient activity recognition using lightweight CNN and DS-GRU network for surveillance applications. *Appl. Soft Comput.* **2021**, *103*, 107102. [CrossRef]
19. Paluru, N.; Dayal, A.; Jenssen, H.B.; Sakinis, T.; Cenkeramaddi, L.R.; Prakash, J.; Yalavarthy, P.K. Anam-Net: Anamorphic depth embedding-based lightweight CNN for segmentation of anomalies in COVID-19 chest CT images. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 932–946. [CrossRef] [PubMed]
20. Peng, X.; Kong, L.; Fuh, J.Y.; Wang, H. A Review of Post-Processing Technologies in Additive Manufacturing. *J. Manuf. Mater. Process.* **2021**, *5*, 38. [CrossRef]
21. Lane, B.; Moylan, S.; Whitenton, E. Post-process machining of additive manufactured stainless steel. In Proceedings of the 2015 ASPE Spring Topical Meeting: Achieving Precision Tolerances in Additive Manufacturing, Raleigh, NC, USA, 27 April 2015.
22. Mishra, P.; Sood, S.; Pandit, M.; Khanna, P. Additive Manufacturing: Post Processing Methods and Challenges. *Adv. Eng. Forum* **2021**, *39*, 21–42. [CrossRef]
23. 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. [CrossRef]
24. 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), Boston, MA, USA, 20–23August 2023; IDETC-CIE2023; American Society of Mechanical Engineers: New York, NY, USA, 2023. [CrossRef]
25. Symbiotic Human-Robot Solutions for Complex Surface Finishing Operations. Available online: https://cordis.europa.eu/project/id/637080 (accessed on 13 December 2023).
26. 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), Kyoto City, Japan, 4–7 July 2022; IEEE: Piscataway, NJ, USA, 2022. [CrossRef]

27. 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), Chennai, India, 4–5 May 2017; IEEE: Piscataway, NJ, USA, 2017. [CrossRef]
28. 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, Dublin, Ireland, 25–27 October 2012; IEEE: Piscataway, NJ, USA, 2012. [CrossRef]
29. A Recipe for Training Neural Networks. Available online: https://karpathy.github.io/2019/04/25/recipe/ (accessed on 19 December 2023).
30. Chaoub, A.; Cerisara, C.; Voisin, A.; Iung, B. Deep Learning Representation Pre-Training for Industry 4.0. *PHM Soc. Eur. Conf.* **2022**, *7*, 571–573. [CrossRef]
31. 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: Hershey, PA, USA, 2021; pp. 318–339. [CrossRef]
32. 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. [CrossRef]
33. SPADD-Dataset: Early Set of Varied Images Displaying Unpolished Surface Defects on Aluminum. Available online: https://github.com/AlbinEV/SPADD-Dataset (accessed on 13 December 2023).
34. 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), Beijing, China, 15–17 August 2019; IEEE: Piscataway, NJ, USA, 2019. [CrossRef]
35. Mastromichalakis, S. ALReLU: A different approach on Leaky ReLU activation function to improve Neural Networks Performance. *arXiv* **2020**. [CrossRef]
36. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
37. Github, AlexeyAB/Yolo_Mark. Available online: https://github.com/AlexeyAB/Yolo_mark (accessed on 19 December 2023).