

Article

Textile Fabric Defect Detection Using Enhanced Deep Convolutional Neural Network with Safe Human–Robot Collaborative Interaction

Syed Ali Hassan ^{1,2,*} , Michail J. Beliatas ³ , Agnieszka Radziwon ^{3,4}, Arianna Menciassi ^{1,2,5} 
and Calogero Maria Oddo ^{1,2,5,*} 

- ¹ The BioRobotics Institute, Scuola Superiore Sant’Anna, 56026 Pisa, Italy; arianna.menciassi@santannapisa.it
² Department of Excellence in Robotics & AI, Scuola Superiore Sant’Anna, 56026 Pisa, Italy
³ Department of Business Development and Technology, Aarhus University, 7400 Herning, Denmark; mibel@btech.au.dk (M.J.B.); agra@btech.au.dk (A.R.)
⁴ Haas School of Business, University of California Berkeley, 2220 Piedmont Ave., Berkeley, CA 94720, USA
⁵ Interdisciplinary Research Center Health Science, Scuola Superiore Sant’Anna, 56026 Pisa, Italy
* Correspondence: syedal.hassan@santannapisa.it (S.A.H.); calogero.oddo@santannapisa.it (C.M.O.)

Abstract: The emergence of modern robotic technology and artificial intelligence (AI) enables a transformation in the textile sector. Manual fabric defect inspection is time-consuming, error-prone, and labor-intensive. This offers a great possibility for applying more AI-trained automated processes with safe human–robot interaction (HRI) to reduce risks of work accidents and occupational illnesses and enhance the environmental sustainability of the processes. In this experimental study, we developed, implemented, and tested a novel algorithm that detects fabric defects by utilizing enhanced deep convolutional neural networks (DCNNs). The proposed method integrates advanced DCNN architectures to automatically classify and detect 13 different types of fabric defects, such as double-ends, holes, broken ends, etc., ensuring high accuracy and efficiency in the inspection process. The dataset is created through augmentation techniques and a model is fine-tuned on a large dataset of annotated images using transfer learning approaches. The experiment was performed using an anthropomorphic robot that was programmed to move above the fabric. The camera attached to the robot detected defects in the fabric and triggered an alarm. A photoelectric sensor was installed on the conveyor belt and linked to the robot to notify it about an impending fabric. The CNN model architecture was enhanced to increase performance. Experimental findings show that the presented system can detect fabric defects with a 97.49% mean Average Precision (mAP).

Keywords: human–robot interaction (HRI); fabric defect detection; computer vision; safe human–robot collaboration; convolutional neural network; transfer learning; quality inspection; AI; DCNN; artificial intelligence in textiles; computer vision for quality control; industrial robotics for fabric inspection



Citation: Hassan, S.A.; Beliatas, M.J.; Radziwon, A.; Menciassi, A.; Oddo, C.M. Textile Fabric Defect Detection Using Enhanced Deep Convolutional Neural Network with Safe Human–Robot Collaborative Interaction. *Electronics* **2024**, *13*, 4314. <https://doi.org/10.3390/electronics13214314>

Academic Editor: Eva Cernadas

Received: 31 August 2024

Revised: 24 October 2024

Accepted: 29 October 2024

Published: 2 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Motivation and Objectives

The textile sector could tremendously benefit from the implementation of the combination of robotic technologies and artificial intelligence. These could, among others, improve quality control and the speed and flow of the production process through the implementation of industry 4.0 and smart factory concepts [1–6]. The need for effective and precise fabric defect identification is essential to maintain strict quality requirements. To meet demand and boost profits, certain sectors of the textile industry have embraced more digitally enabled production methods. Similarly, other sectors have started to include digital and sustainable solutions in their final product portfolios in response to a growing worldwide demand for sustainability and the principles that surround it [7]. Manual inspection procedures that are labor-intensive and prone to errors are proving ineffective [8].

As a result, the introduction of deep learning approaches, especially deep convolutional neural networks (DCNNs), appears to have potential for automating the detection of fabric defects, thus providing a more accurate and efficient alternative. Fabric defects, such as broken ends, holes, double-ends, and other defects, must be identified during inspection. Manual inspection of fabric is time-consuming, prone to error, and requires relevant human attention. Smart solutions are required to precisely classify and localize the fabric defects in real time. Defective fabric exposes the producer to costs, which is why quality control is a must. Our study shows how the application of modern technologies can make it faster, cheaper, and better while following the principle of HRI, where the alarm sound prompts the quality inspection operators to remove defective fabrics for further inspection in a human–AI robot collaborative manner. The authors did not consider the cost as a major constraint because the whole system is developed in a research laboratory, and it is not intended to be a product currently.

1.2. State of the Art and Related Work

Numerous studies in a variety of areas, including healthcare imaging, object recognition, and manufacturing quality control, have demonstrated the prowess of DCNN in complex pattern recognition tasks [9–12]. However, the use of advanced technologies, such as enhanced DCNNs, must be supported by a strategic assessment of the human–robot interaction (HRI) dynamics in the production environment. The introduction of collaborative robotics has tremendous potential for increasing production volumes and quality while, in parallel, protecting human workers [13]. Integrating fabric defect detection with safe HRI is a holistic approach to textile manufacturing that addresses both the technical and human-centric components of the industry [14].

More recently, the development of transfer learning methodologies has allowed neural network models that have already been trained to be applied to new applications and tasks [15]. Such a technique alleviates the need for a suitably big pool of annotated data and has been found to be useful [16–19], thus enabling precise defect detectors [20,21]. Zhen et al. utilized transfer learning [22] for the identification of fabric defects. Their system is based on the YOLOv8 (You Only Look Once) [23] object detector, which is well-known for its high speed and accuracy.

Automatic fabric identification algorithms have been the subject of years of research and development, with some promising techniques already published [24]. The speed and accuracy of automatic fabric defect detection have always been limited by different background textures and variations in the production environments of businesses. Experts and researchers have undertaken research in this sector in recent years, and various techniques for the detection of fabric have been developed [25]. The three most common types of these approaches are model analysis [26,27], spectrum analysis [28,29], and DL techniques [30–32], which are compared in Table 1. DL-based algorithms have substantially outperformed them in terms of recognition speed and precision, and they have started to be used on a broad scale for industrial fabric defect identification.

Table 1. Merits and drawbacks of approaches used for fabric defect detection.

Approach	Methods	Merits	Drawbacks
Model Analysis	Gaussian Markov Random Field (GMRF)	Mostly utilized for classification/segmentation problems	Hard to assess performance
Spectrum Analysis	Wavelet transform	Good for feature extraction of image	Outdated
Deep Learning	CNN-based	High performance	Huge training data and manual annotation needed

The recognition of objects is one of the most essential jobs in computer vision, and it is often employed for defect detection. The aim is to acquire precise details regarding the location and classification of objects. DL-based object detection techniques are currently developing quite quickly. DL-based defect detection networks can be broadly classified into

two types. The first type is one-stage networks, known as an SSD (single-shot detector) [33]. The second-stage networks are known as Faster R-CNN (Region-CNN) [32].

Single-shot detectors may operate on low-power devices like tablets and phones, but region-based detectors require significant computational power. Emerging techniques and methods for identifying fabric problems have been created in the last few years due to the extensive industrial use of visual analysis and computer vision technologies. Jamali and colleagues proposed [34] to create a diagnostic process for identifying and categorizing warp and weft problems using a technique of image processing. In [35], 276 images connected to the textile materials were gathered, prepared, and labeled for publication. Following this process, CNN was utilized to develop algorithms for fabric detection; only three categories of defects—belt yarn, knot tying, and holes—were identified. The authors in [36] discussed the difficulties associated with quality assurance in industrial environments, concentrating on the examination and rectification of metallic surfaces to detect manufacturing defects. The method uses a multi-agent robotic mechanism, which consists of a handler agent to manipulate the object, an inspection agent to see potential faults in the object's 3D structure, and a grinder agent to fix surface abnormalities.

The identification technique is crucial for human–robot collaboration-based defect identification, and visual assessment is a popular way to detect surface defects [37]. Traditionally, ML techniques like support vector machine (SVM) have been applied extensively. A multimodal fusion technique for welding flaws with SVM was presented in [38]. Generalized finished local binary structures for metal surface flaws were introduced in the study by Luo and colleagues [39]. For steel surface flaws, Xiao et al. [40] also suggested an SVM with kernels. However, the models may not perform as intended in traditional ML techniques, which heavily rely on explicit feature extraction. These days, employing DL for defect assessment has grown popular because of advancements in artificial intelligence [41]. In contrast to conventional machine learning techniques, DL approaches enable the automatic extraction of fault features through a training process. It makes identification more general and eliminates the need for explicit feature extraction. To extract the unique characteristic of steel surface flaws, Gao et al. [42] suggested CNN, and this technique was successfully created in a real-world scenario. A concurrent CNN with various image sizes was proposed by Liu et al. [43]. A DL technique for steel surface flaws was presented by Hao et al. [44]. A knowledge-augmented wide learning approach for wafer flaws was presented by Wang et al. [45]. A triplet-graph logic system with a fault segmentation was introduced by Bao et al. [46], which distinguishes between fault and non-fault regions. To identify small defects, Zhang et al. [47] suggested a weakly supervised technique that utilized a category-aware component. In this paper, a DL-based technique is utilized for the detection of defects in fabric, such as broken ends, coffee stains, holes, double-ends, ink stains, double picks, stains, missing picks, oil stains, pin marks, tiny holes, slip knots, and thread outs, through safe human–robot collaboration. The algorithm is written and enhanced to alert the human regarding detected fabric defects in real time, with 97.49% mAP. The block diagram of the implementation can be seen in Figure 1, and the flowchart of the robot program is shown in Figure 2. The actual experimental setup is shown in Figure 3.

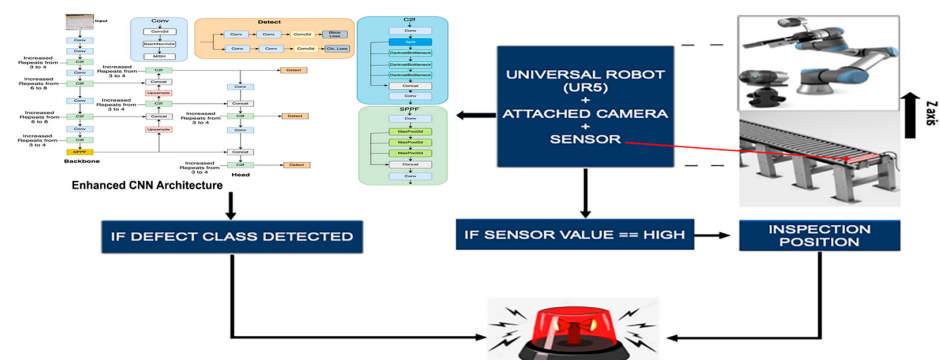


Figure 1. Block diagram of the system.

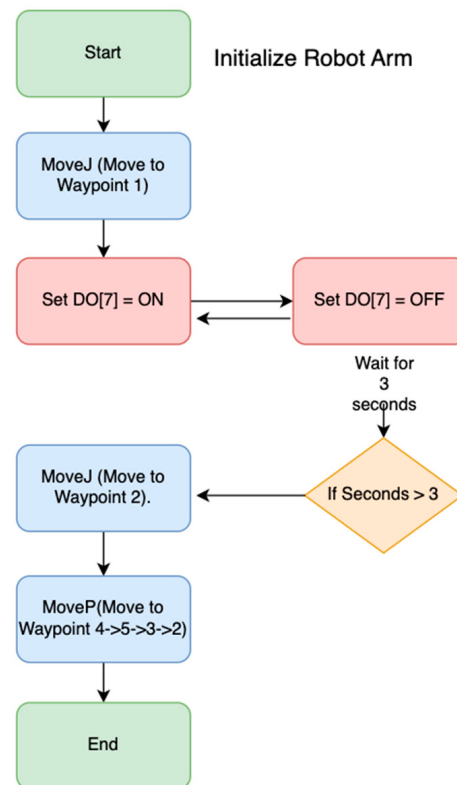


Figure 2. Flowchart of robot program.

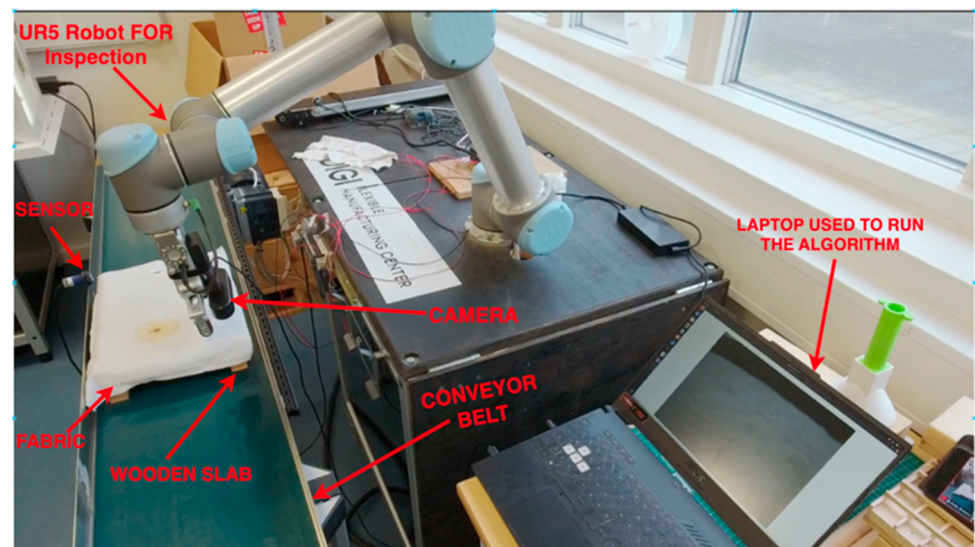


Figure 3. The experimental setup of the defect detection system.

1.3. Contribution of This Study

While these techniques work well for defect detection, it is unlikely that they will be immediately applied to human–robot collaboration-based defect recognition. The role of human verification is crucial for improving human–robot collaboration-based fabric defect detection. A human–robot collaboration policy for reviewing is required so that humans are able to verify the defective fabrics; otherwise, they would have to commit to the cumbersome task of rechecking every sample. For this reason, an alarm-based human–robot collaboration system is introduced in this paper for real-time fabric defect detection. In this study, YOLOv8n architecture is enhanced and used as a foundation

model from the single-shot detector group, where n means nano because it consists of fewer layers compared to the large model of YOLOv8L. The reason for using the nano version as a foundation model is its ability to run on low-power devices without any need for powerful GPUs.

The paper is organized as follows. Section 2 offers the description of the experimental setup, the robot control, and the developed and applied algorithm. Section 3 describes the process of dataset generation, the software environment, the architecture enhancements, training, and the results of the experiment. Finally, Section 4 concludes the paper and proposes a future research outlook.

2. Materials and Methods

2.1. Experimental Setup and Robot Control

The developed platform for inspection of textiles consisted of an anthropomorphic robot arm (Universal Robot UR5, Odense, Denmark) with an attached camera (Logitech c310, Ecublens, Switzerland). The selected robot is a collaborative robot that enables safe human–robot interaction during experimental activities. A conveyor belt (Rollco, Vejle, Denmark) with a photoelectric sensor (CY-121B-P, Panasonic, Osaka, Japan) was connected to a Universal Robot arm and a laptop running the CNN. The enhanced CNN was trained on multiple classes for fabric defect detection, which is used in this experiment. A more detailed description of the experimental setup is discussed in the following paragraph.

The anthropomorphic robot is a lightweight collaborative machine with a 850 mm reach, 6 rotating joints, and a 5 kg payload. Using a proprietary communication protocol and an open Ethernet connection, the robot was linked to a laptop executing the algorithm. The controller was implemented on a machine with Intel i7 3.2 GHz processor and 3 GB RAM running Linux 2.6.38 (Universal Robots, Odense, Denmark). The robot is capable of functioning as a collaborative robot [48] because it satisfies EN ISO 10218-1:2006 standard point 5.10.5, guaranteeing that a collision would not have severe consequences. Thus, the robot can be used for inspection purposes because safety guards around persons and the robot are not necessary. One of the built-in safety features is the ability to halt when the torque at the robot joint varies from what is predicted using the factory settings reported hereafter. A precautionary stop is created if the torque outpaces 42 Nm for the first three (big) joints and 10 Nm for the final three (tiny) joints. The torque at every joint is estimated in real time utilizing a model of the robot. A preventive stop is also created if the joint velocity reaches 3.2 rad/s or the outside force exceeds 150 N. The anthropomorphic robot is programmed to explore the fabric, which comes by means of the conveyor belt. The camera was attached to the robot end effector and connected with the laptop (Intel i7, 32 GB RAM) equipped with a GPU (ASUS RTX 3060, 6 GB). The robot was programmed so as to keep a 32 mm nominal distance between the camera and the textile sample. The cylindrical photoelectric sensor attached to the conveyor belt (Rollco.dk), with a 100 mm sensing range and a 1 ms response time, was responsible for informing the robot about upcoming fabric so as to trigger the robot to start the inspection task. If the algorithm detects fabric defects, an alarm is raised to inform the operator to remove the fabric; otherwise, the next fabric arrives. The vision algorithm and the robot controlling program are working separately. It is a system based on human–robot interaction where the vision algorithm, which is running on a laptop, is responsible for defect detection and alarm alert and the robot program is responsible for moving a robot hand above the fabric to provide the best positions for the camera for detection of defects and to raise the alarm. Once the alarm is raised, then it is the responsibility of humans to verify the fabric. The current system is based on human–robot interaction. Our deliberate choice was to involve humans to verify the detected fabric. This makes the system semi-autonomous, as the robotic arm is not grasping the defective fabric. If we add grasping, it would remove the human-verification element. As our priority is to reduce human effort in checking all of the fabric defects, instead of replacing the human. To move the robot for inspection, MoveJ and MoveP commands are used, as shown in Algorithm 1. MoveJ is a motion command to move robot joints to predefined waypoints.

This command can provide faster movements, but it may not always follow a straight path. MoveP is a pose-based movement that is more precise, and it was used to move the robot hand on each part of the fabric.

Algorithm 1 Robot program

```
//Initialization of the program
Initialize robot arm
Movej (Joint-based movement)
Begin Movej
Move to Waypoint_1
Set Digital Output DO [7] to ON (Received from sensor)
Wait for 3 s
Set Digital Output DO [7] to OFF
Move to Waypoint_2 (another joint position)
End Movej

MoveP (Pose-based movement)
Begin Movep
Move to Waypoint_4 (predefined position)
Move to Waypoint_5
Move to Waypoint_3
Move to Waypoint_2
End MoveP
```

2.2. AI Methods for Sensor Data Elaboration and Software Environment

The code to elaborate sensor data was edited in Python programming language, and an implementation of a real-time object detection and alerting system is described in Algorithm 2. The method detects objects using an enhanced deep CNN model. The main goal is to detect fabric defects in real time using a camera and to generate a sound alarm (implemented with Pygame library) when they are detected, including the design of a cooldown technique to minimize continual alarms. The YOLO model is loaded, and the camera is configured for object detection. Object classes, such as broken ends, coffee stains, holes, double-ends, ink stains, double picks, stains, missing picks, oil stains, pin marks, tiny holes, slip knots, and thread outs, are trained for real-time detection.

Algorithm 2 Pseudocode code of the alarm system

```
# Initialize alarm state
alarm_triggered = False
cooldown_duration = 10
cooldown_start_time = 0

# Main processing loop

while True:
    frame, detection_result = capture_and_detect()
    for obj in detection_result:
        confidence, obj_class = process_detection(obj)
        if confidence > 50:
            Draw_and_display(frame,obj,classnames[obj,class],confidence)

    if not alarm_triggered:
        trigger_alarm()
        cooldown_check_and_reset()
        show_frame(frame)
```

The primary processing loop is the core of the system. The system captures a frame and uses the object detection model for each iteration. After processing, if the confidence of the detected objects is higher than a predetermined threshold, relevant information is shown on the frame. Controlling the system's response to detections can be achieved by setting a threshold. While a lower threshold might catch more identifications, it could raise the risk of false positives, while a higher threshold could reduce false positives, but it could also lead to many true detections being missed. Several confidence levels have been tested during development, and a 50% level was found to be best. The 60% confidence level is also good, but it was missing some defects, and it had an accuracy of less than 50% but more than 45%. This could be explained by our training, which also entailed tiny defects. The goal of our system is to alert the human about all defects to be further human verified. The cooldown method assists in avoiding the overabundance of alerts for repeated or same detections that the system could otherwise overload users with. Introducing a predetermined delay between successive warnings enhances system usability and reduces the possibility of disturbing users with frequent alerts. An alert is activated upon detection, and a cooldown mechanism stops continual alarms for a set period of time. The operations for object detection and warning are reported in the pseudocode. The experimental environment was established using PyTorch 1.8 as the framework for DL and Ubuntu 20.04 as an operating system to evaluate the effectiveness of the proposed approach. The latest version of the YOLO model (v8) served as the baseline model.

3. Experimental Protocol, Results, Architecture, and Enhancement

This section covers all aspects of creating and modifying the datasets as well as the outcomes that the detection model produced with the enhancement in the architecture. All studies used a consistent set of hyperparameters during the training phase (Table 2).

Table 2. Hyperparameters used in training.

Hyperparameters	Value
Image Size	640 by 640
Learning Rate	0.01
Optimizer	SGD
Batch Size	32
Weight Decay	0.0005
Momentum	0.937
Epochs	100

3.1. Dataset Creation and Augmentation

In total, 13 classes of textile defects were identified, collected, and used for this experiment. These consist of textiles, characterized by broken ends, coffee stains, medium to large holes, double-ends, ink stains, double picks, stains, missing picks, oil stains, pin marks, tiny holes, slip knots, and thread outs. These classes represent a selection of the most common defects in fabrics, and they were selected as a proof of concept for our experiments to evaluate whether the presented apparatus and AI methods allow for successfully detecting the defects varying in shape, size, texture, and color.

Several videos were recorded, and frames were extracted. Each frame was inspected for high quality and annotated. All selected images were annotated manually, and augmentation techniques, such as brightness level (−15% to 15%), noise, blur (1%), rotation (−10 to 10 degrees), saturation (15%), and flip (horizontal, vertical), were applied. Then, 100 images for each class were prepared before augmentation and then increased to 3× after augmentation. Overall, 3900 RGB (red, green, blue) images were selected for the 13 defect classes, of which 65% of the images were used in training, 15% for testing, and 20% for validation. To ensure the model learned classes correctly, we examined every annotated image prior to training after completing the class annotation. The dataset generation process was

tough and tedious because we methodically created the dataset and reviewed it numerous times to prevent undesirable outcomes throughout the training phase. Moreover, the mAP of the model would have been decreased if the class name had been labeled improperly, necessitating careful attention to annotate thousands of images. A comparison of image distribution before and after augmentation for defect classes is plotted in Figure 4.

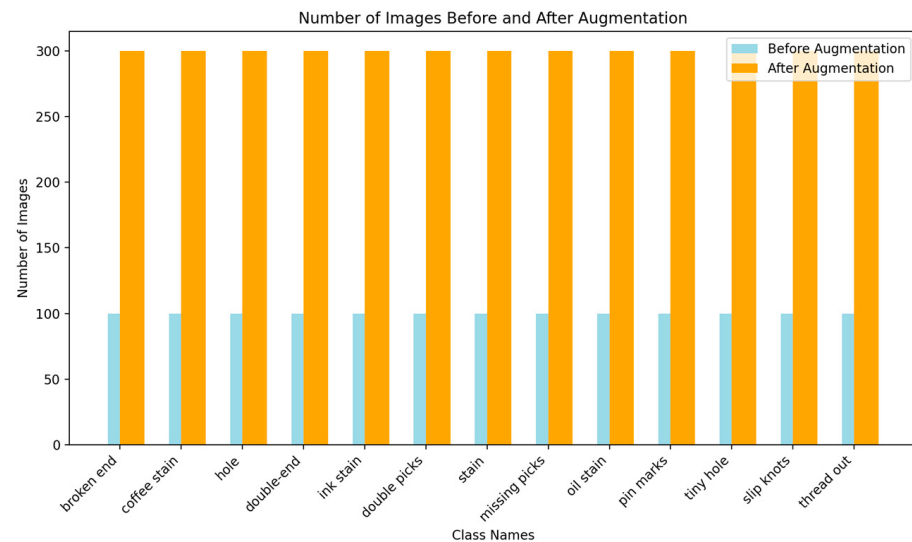


Figure 4. Comparison of Image distribution before and after augmentation for defect classes.

3.2. Defect Identification Training and Experimental Results

The CNN model was trained for 100 epochs, and the trained model was subsequently evaluated utilizing samples that had not been utilized during training. Transfer learning approaches [49–52] were used to train all of the 13 classes. One of the most recent YOLOv8n [53] architectures was enhanced and then chosen as the backbone architecture because of its fast execution time and adequate mAP. After training with an enhanced model on the self-created dataset, the achieved mAP was 97.49%, the recall was 98.45%, the precision was 91.55%, and the inference time was 11.36ms. CNN performed excellently in recognizing defects in the fabric. The trained model demonstrated excellent mAP in differentiating between broken ends, coffee stains, holes, double-ends, ink stains, double picks, stains, missing picks, oil stains, pin marks, tiny holes, slip knots, and thread outs. By incorporating deep learning techniques, fabric images could be thoroughly analyzed, which made it possible for the system to adjust to a variety of defective features. In [54], the foundational model for detecting fabric defects is the RefineDet neural network, which attained 82.6% mAP. The model trained in this paper achieved 97.49% mAP for fabric defect detection, which is better than the model trained with a backbone of RefineDet. The project’s ability to detect fabric defects in real time is one of its noteworthy accomplishments. The confusion matrix of the enhanced model can be seen in Figure 5, which highlights the accuracy of 13 types of fabric defects. The diagonal elements in the illustration represent the detection rates, rows represent labels, and columns display classes.

The matrix is used to evaluate both the correct predictions (true positives) and the errors (false positives/false negatives) in 13 defect classes and the background. The model gives great results for most of the defect types, with defects, such as “broken-end”, “coffee-stain”, etc., perfect at 1.00. There are certain categories, like ink stains, missing picks, and tiny holes, which are correlated visually; this makes their accuracy slightly lower, but it is acceptable for these categories. In the future, a large dataset can be created to increase the accuracy of challenging defect categories. Figures 6 and 7 show the comparison of F1 confidence where the enhanced model achieved a better score. As shown in Figures 8 and 9, our model attains excellent accuracy in detecting every defect and in the detection outcomes. The accuracy evaluation metrics used in the present research were mAP, recall (R), and

precision (P). To be more precise, P is the ratio of the predicted detection region to the true detection space, and R is the percentage of correctly predicted classes to the entire number of necessary classes. The accuracy of the entire sample for which the predicted box is greater than 50% of the actual box is determined through mAP. A greater mAP score corresponds to a higher prediction accuracy. Furthermore, the study employed network parameter volume and inference time as performance metrics. Reduced model size denotes lower memory consumption, whereas shorter inference times imply greater real-time performance. N stands for all samples; Q stands for discovered defect classes in this study; TP stands for true positive data; FP stands for false positive data; and N stands for the total quantity of samples. The i -th class mean precision can be expressed as AP_i . The calculation formula is shown in the equations below.

$$AP_i = \frac{TP}{TP + FP} \quad (1)$$

$$mAP = \frac{\sum_{i=1}^Q AP_i}{Q} \times 100\% \quad (2)$$

$$P = \frac{TP}{TP + FP} \quad (3)$$

$$R = \frac{TP}{P} \quad (4)$$

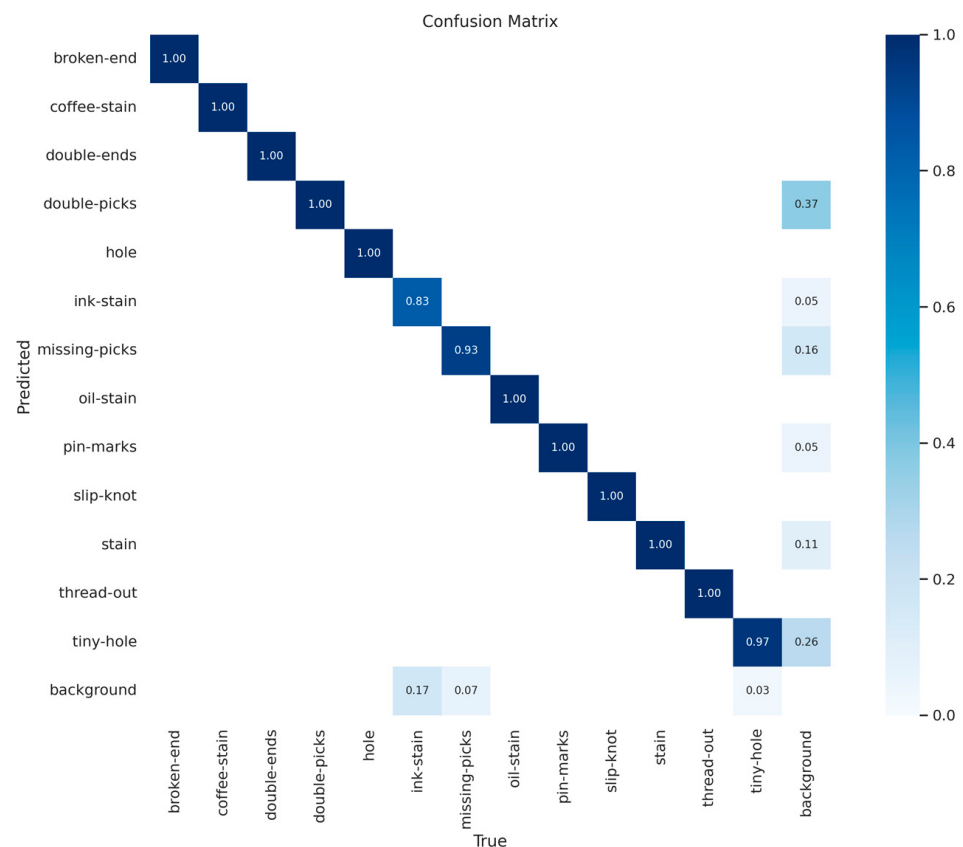


Figure 5. Confusion matrix of the enhanced trained model.

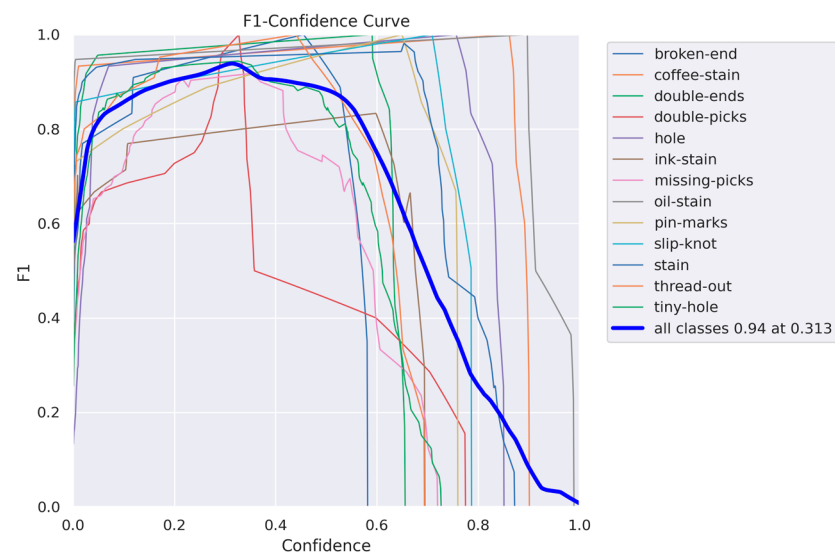


Figure 6. F1—confidence Curve of the enhanced model.

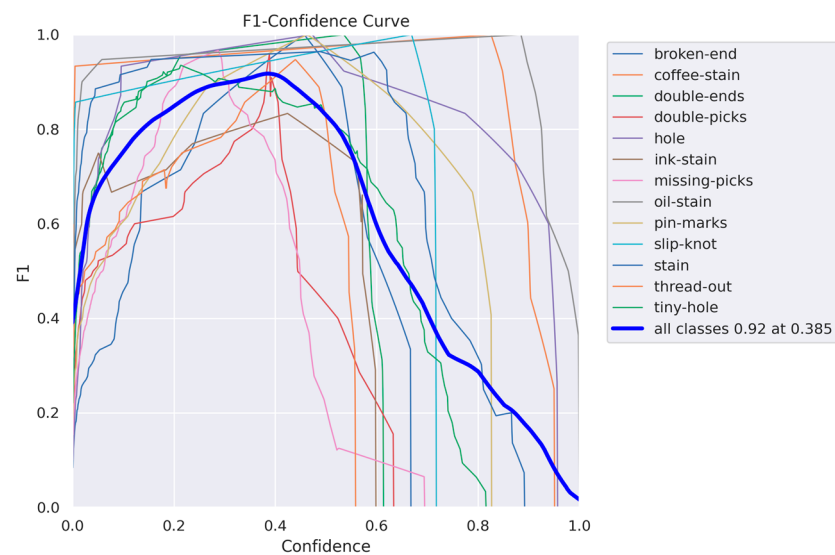


Figure 7. F1—confidence Curve of the default model.

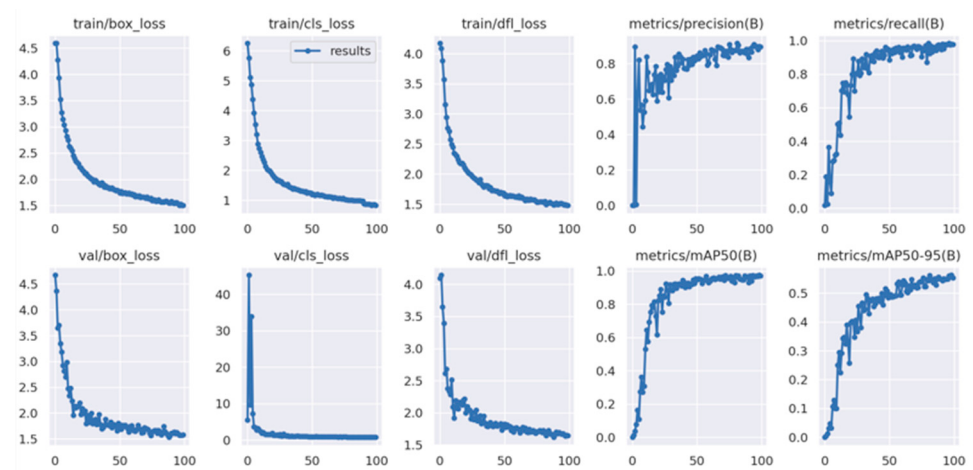


Figure 8. Results after training with enhanced model.



Figure 9. Real-time detection experiments conducted using the reported model.

3.3. Network Architecture of CNN

The CNN model architecture has three main components: the backbone, the neck, and the head. In this model, the backbone is responsible for extracting features from the input image via a sequence of convolutional layers, C2f blocks, and a Spatial Pyramid Pooling Fast (SPPF) block. The neck aims at enhancing multi-scale feature representation through convolutional layers, concatenation, and upsampling, which aggregate features from different levels of the backbone. The detection layers in the head compute bounding box regression and classification loss to make final predictions. The key constituents are Conv layers with Conv2d, BatchNorm2d, and SiLU activation; C2f blocks with DarknetBottleneck structures; and SPPF for multi-scale pooling. Accordingly, the model is modularly designed such that it can detect objects efficiently and accurately in images, and it is therefore useful in real-time object detection applications. The network architecture of the CNN model is shown in Figure 10.

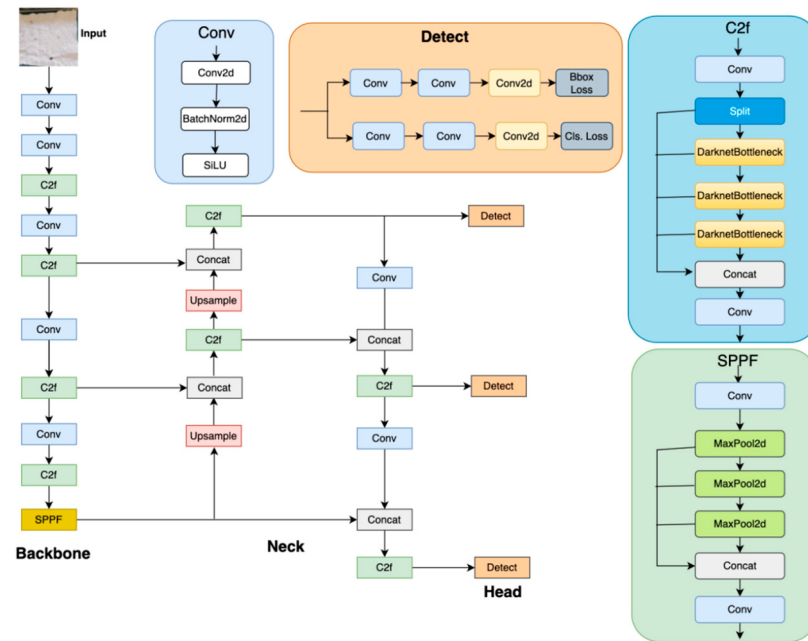


Figure 10. Network architecture of the CNN model.

3.4. Enhanced Network Architecture of CNN

To increase the model's performance, several changes were implemented. The model width and the depth multipliers both increased from 0.50 to 0.75. This modification allows for more layers within each module and a higher number of convolutional filters that enhance feature extraction and learning capabilities of the model. The backbone of the model consists of a series of convolutional layers and cross stage partial with full connection (C2F) modules; the number of repeats in the C2F modules increased for deeper feature extraction. The head of the model is responsible for aggregating features and making detections using the same technique; the repeats are increased in the C2F modules. The SiLU activation is replaced with the mish [55] activation function for deep propagation. These modifications enable the model to learn more abstract and detailed features, enhance its ability to handle more complex and varied data, and enhance its robustness by reducing overfitting. After enhancement, the model mAP increased from 97.26% to 97.49%, the precision increased from 90.05% to 91.55%, the recall increased from 96.68% to 98.45%, and the inference time increased from 4.2 ms to 11.36 ms, which is acceptable. This minor increase in inference time does not prevent practical use of the CNN for textile defect detection. These results confirm that enhancing the architecture further not only increases its precision, recall, and mAP but also its inference time. The enhanced network architecture of the CNN model is shown in Figure 11, and the results comparison of both models can be seen in Table 3.

Table 3. Result comparison of both models.

Model	mAP%	Recall%	Inference (ms)	Precision%	F1%
Default CNN	97.26	96.68	4.2	90.05	92.00
Enhanced CNN	97.49	98.45	11.36	91.55	94.00

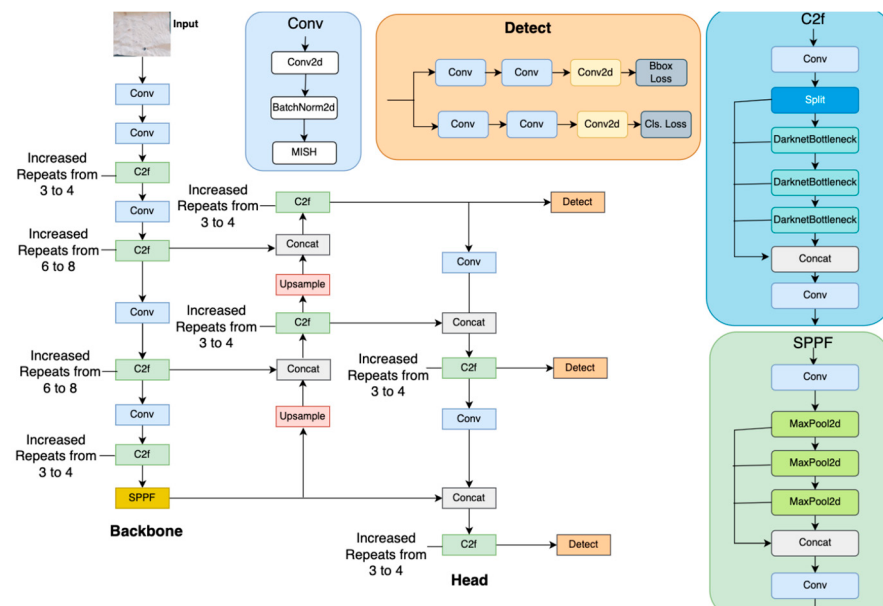


Figure 11. Enhanced network architecture of CNN model.

4. Discussion and Conclusions

In the context of safe human–robot collaboration, this study introduced an innovative method to handle issues in fabric inspection by utilizing modern technology, specifically robotic systems, cameras, and deep convolutional neural networks (DCNNs). The enhanced method shows 97.49% mAP in performing the classification and detection of different types of fabric defects. An orderly and effective inspection procedure is guaranteed by the use of a Universal Robot (UR5) equipped with a camera and a photoelectric sensor on a conveyor belt. The achieved mAP and inference time highlight the potential of the system for useful applications in the textile sector. This minor increase in inference time does not prevent practical use of CNN for textile defect detection. This research adds to continuous endeavors within the textile sector to improve production efficiency and quality control through the implementation of artificial intelligence and human–robot interaction. The CNN’s minimal latency performance guarantees early defect detection during the production process. Rapid defect detection and action are made possible by the neural network’s effective image processing, which enables a seamless connection with the Universal Robot. Potential for human–robot collaboration are made possible by the system’s collaborative character, which combines the efforts of the Universal Robot and the CNN. The purpose of this system is not to replace humans but to help the operator who is performing fabric inspections. In the textile industry, it is not easy for humans to accurately check each fabric’s quality when there are thousands of fabrics waiting for inspection. This system can help the textile industry to avoid human errors and enable them to increase their production time while reducing accidents at work, as well. Operators can take advantage of the system’s real-time alert and take appropriate action as needed. Future efforts can be targeted at creating a larger fabric dataset and further improving the system through implementation in a real operational scenario within the textile industry. The system can be modified for a number of uses, including quality assurance and surveillance, which advance the fields of computer vision and intelligent systems. To improve the collaboration, the inclusion of feedback in the algorithm after the operator reviews the defect can be implemented so that it can be further specialized for the specific products of the corresponding textile fabric, and a database can be created to create an analytical portal for the system.

Author Contributions: Conceptualization, S.A.H., M.J.B. and C.M.O.; methodology, S.A.H., M.J.B. and C.M.O.; formal analysis, M.J.B., C.M.O., A.R. and S.A.H.; investigation, S.A.H. and M.J.B.; dataset

creation, S.A.H.; writing—review and editing, S.A.H., M.J.B., A.R. and C.M.O.; supervision, C.M.O., M.J.B., A.R. and A.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research has received funding from the EU’s Horizon2020 research and innovation program under grant agreement no. 956745, EINST4INE: The European Training Network for Industry Digital Transformation Across Innovation Ecosystems. The content of this publication does not reflect the official opinion of the EU. Responsibility for the information and views expressed in the publication lies entirely with the authors.

Data Availability Statement: The data presented in this study are available upon request from the corresponding authors.

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

References

1. Felsberger, A.; Qaiser, F.H.; Choudhary, A.; Reiner, G. The impact of Industry 4.0 on the reconciliation of dynamic capabilities: Evidence from the European manufacturing industries. *Prod. Plan. Control* **2022**, *33*, 277–300. [\[CrossRef\]](#)
2. Radziwon, A.; Bilberg, A.; Bogers, M.; Madsen, E.S. The smart factory: Exploring adaptive and flexible manufacturing solutions. *Procedia Eng.* **2014**, *69*, 1184–1190. [\[CrossRef\]](#)
3. Gloy, Y.-S. *Industry 4.0 in Textile Production*; Springer: Berlin/Heidelberg, Germany, 2021.
4. Forno, A.J.D.; Bataglini, W.V.; Steffens, F.; de Souza, A.A.U. Industry 4.0 in textile and apparel sector: A systematic literature review. *Res. J. Text. Appar.* **2023**, *27*, 95–117. [\[CrossRef\]](#)
5. Weiß, M.; Tilebein, M.; Gebhardt, R.; Barteld, M. Smart Factory Modelling for SME: Modelling the Textile Factory of the Future. In Proceedings of the Business Modeling and Software Design: 8th International Symposium, BMSD 2018, Vienna, Austria, 2–4 July 2018; Proceedings 8. pp. 328–337.
6. Singh, P.; Karpavice, J.; Beliatas, M.; Presser, M. *Digital Dataspace and Business Ecosystem Growth for Industrial Roll-to-Roll Label Printing Manufacturing: A Case Study*; Thinkmind Digital Library: Lisbon, Portugal, 2023.
7. Joergens, C. Ethical fashion: Myth or future trend? *J. Fash. Mark. Manag. Int. J.* **2006**, *10*, 360–371. [\[CrossRef\]](#)
8. Ngan, H.Y.T.; Pang, G.K.H.; Yung, N.H.C. Automated fabric defect detection—A review. *Image Vis. Comput.* **2011**, *29*, 442–458. [\[CrossRef\]](#)
9. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#)
10. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1–9. [\[CrossRef\]](#)
11. Esteva, A.; Kuprel, B.; Novoa, R.A.; Ko, J.; Swetter, S.M.; Blau, H.M.; Thrun, S. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **2017**, *542*, 115–118. [\[CrossRef\]](#)
12. Lystbæk, M.; Giannakidis, A.; Beliatas, M.J.; Olsen, M. Removing Unwanted Text from Architectural Images with Multi-Scale Deformable Attention-Based Machine Learning. In Proceedings of the 2023 IEEE International Conference on Imaging Systems and Techniques (IST), Copenhagen, Denmark, 17–19 October 2023; pp. 1–5.
13. Haddadin, S.; Albu-Schäffer, A.; Hirzinger, G. Safety evaluation of physical human-robot interaction via crash-testing. *Robot. Sci. Syst.* **2007**, *3*, 217–224.
14. Koustoumpardis, P.N.; Chatzilygeroudis, K.I.; Synodinos, A.I.; Aspragathos, N.A. Human robot collaboration for folding fabrics based on force/rgb-d feedback. In Proceedings of the Advances in Robot Design and Intelligent Control: Proceedings of the 24th International Conference on Robotics in Alpe-Adria-Danube Region (RAAD), Bucharest, Romania, 27–29 May 2015; pp. 235–243.
15. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *J. Big Data* **2016**, *3*, 1345–1459. [\[CrossRef\]](#)
16. Krichen, M. Convolutional Neural Networks: A Survey. *Computers* **2023**, *12*, 151. [\[CrossRef\]](#)
17. Ng, W.; Minasny, B.; McBratney, A. Convolutional neural network for soil microplastic contamination screening using infrared spectroscopy. *Sci. Total Environ.* **2020**, *702*, 134723. [\[CrossRef\]](#)
18. Hassan, S.A.; Khalil, M.A.; Auletta, F.; Filosa, M.; Camboni, D.; Menciassi, A.; Oddo, C.M. Contamination Detection Using a Deep Convolutional Neural Network with Safe Machine—Environment Interaction. *Electronics* **2023**, *12*, 4260. [\[CrossRef\]](#)
19. Jia, Z.; Shi, Z.; Quan, Z.; Shunqi, M. Fabric defect detection based on transfer learning and improved Faster R-CNN. *J. Eng. Fiber. Fabr.* **2022**, *17*, 15589250221086648. [\[CrossRef\]](#)
20. Gopalakrishnan, K.; Khaitan, S.K.; Choudhary, A.; Agrawal, A. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Constr. Build. Mater.* **2017**, *157*, 322–330. [\[CrossRef\]](#)
21. Burra, L.R.; Karuna, A.; Tumma, S.; Marlapalli, K.; Tumuluru, P. MobileNetV2-based Transfer Learning Model with Edge Computing for Automatic Fabric Defect Detection. *J. Sci. Ind. Res.* **2022**, *82*, 128–134.
22. Cheng, Z.; Luo, X.; Shi, Y.; Kita, K. Fabric defect detection algorithm based on YOLOv3 Transfer learning. In Proceedings of the International Conference on Frontiers of Electronics, Information and Computation Technologies, Changsha, China, 21–23 May 2021; pp. 1–7.

23. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788. [\[CrossRef\]](#)
24. Jeyaraj, P.R.; Nadar, E.R.S. Computer vision for automatic detection and classification of fabric defect employing deep learning algorithm. *Int. J. Cloth. Sci. Technol.* **2019**, *31*, 510–521. [\[CrossRef\]](#)
25. Yue, X.; Wang, Q.; He, L.; Li, Y.; Tang, D. Research on tiny target detection technology of fabric defects based on improved Yolo. *Appl. Sci.* **2022**, *12*, 6823. [\[CrossRef\]](#)
26. Cohen, F.S.; Fan, Z.; Attali, S. Automated inspection of textile fabrics using textural models. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 803–808. [\[CrossRef\]](#)
27. Xu, Y.; Meng, F.; Wang, L.; Zhang, M.; Wu, C. Fabric surface defect detection based on GMRF Model. In Proceedings of the 2021 2nd International Conference on Artificial Intelligence and Information Systems, Chongqing, China, 28–30 May 2021; pp. 1–4.
28. Ngan, H.Y.T.; Pang, G.K.H.; Yung, S.-P.; Ng, M.K. Wavelet based methods on patterned fabric defect detection. *Pattern Recognit.* **2005**, *38*, 559–576. [\[CrossRef\]](#)
29. Bianconi, F.; Fernández, A. Evaluation of the effects of Gabor filter parameters on texture classification. *Pattern Recognit.* **2007**, *40*, 3325–3335. [\[CrossRef\]](#)
30. Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
31. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
32. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [\[CrossRef\]](#) [\[PubMed\]](#)
33. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14. pp. 21–37.
34. Jmali, M.; Zitouni, B.; Sakli, F. Fabrics defects detecting using image processing and neural networks. In Proceedings of the 2014 Information and Communication Technologies Innovation and Application (ICTIA), Sousse, Tunisia, 6–8 March 2014; pp. 1–6.
35. Zhang, H.; Zhang, L.; Li, P.; Gu, D. Yarn-dyed fabric defect detection with YOLOV2 based on deep convolution neural networks. In Proceedings of the 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), Enshi, China, 25–27 May 2018; pp. 170–174.
36. Czimmermann, T.; Chiurazzi, M.; Milazzo, M.; Roccella, S.; Barbieri, M.; Dario, P.; Oddo, C.M.; Ciuti, G. An autonomous robotic platform for manipulation and inspection of metallic Surfaces in Industry 4.0. *IEEE Trans. Autom. Sci. Eng.* **2021**, *19*, 1691–1706. [\[CrossRef\]](#)
37. He, Y.; Song, K.; Meng, Q.; Yan, Y. An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Trans. Instrum. Meas.* **2019**, *69*, 1493–1504. [\[CrossRef\]](#)
38. Xiao, M.; Jiang, M.; Li, G.; Xie, L.; Yi, L. An evolutionary classifier for steel surface defects with small sample set EURASIP. *J. Image Video Process* **2017**, *48*, 2017.
39. Tan, A.; Zhou, G.; He, M. Surface defect identification of Citrus based on KF-2D-Renyi and ABC-SVM. *Multimed. Tools Appl.* **2021**, *80*, 9109–9136. [\[CrossRef\]](#)
40. Wang, Y.; Sun, W.; Jin, J.; Kong, Z.; Yue, X. MVGCN: Multi-View Graph Convolutional Neural Network for Surface Defect Identification Using Three-Dimensional Point Cloud. *J. Manuf. Sci. Eng.* **2023**, *145*, 31004. [\[CrossRef\]](#)
41. Liu, Y.; Yuan, Y.; Balta, C.; Liu, J. A light-weight deep-learning model with multi-scale features for steel surface defect classification. *Materials* **2020**, *13*, 4629. [\[CrossRef\]](#)
42. Gao, Y.; Gao, L.; Li, X. A hierarchical training-convolutional neural network with feature alignment for steel surface defect recognition. *Robot. Comput. Integr. Manuf.* **2023**, *81*, 102507. [\[CrossRef\]](#)
43. Surana, A.; Lynch, M.E.; Nassar, A.R.; Ojard, G.C.; Fisher, B.A.; Corbin, D.; Overdorff, R. Flaw Detection in Multi-Laser Powder Bed Fusion Using In Situ Coaxial Multi-Spectral Sensing and Deep Learning. *J. Manuf. Sci. Eng.* **2023**, *145*, 51005. [\[CrossRef\]](#)
44. Hao, R.; Lu, B.; Cheng, Y.; Li, X.; Huang, B. A steel surface defect inspection approach towards smart industrial monitoring. *J. Intell. Manuf.* **2021**, *32*, 1833–1843. [\[CrossRef\]](#)
45. Wang, J.; Gao, P.; Zhang, J.; Lu, C.; Shen, B. Knowledge augmented broad learning system for computer vision based mixed-type defect detection in semiconductor manufacturing. *Robot. Comput. Integr. Manuf.* **2023**, *81*, 102513. [\[CrossRef\]](#)
46. Zhang, J.; Su, H.; Zou, W.; Gong, X.; Zhang, Z.; Shen, F. CADN: A weakly supervised learning-based category-aware object detection network for surface defect detection. *Pattern Recognit.* **2021**, *109*, 107571. [\[CrossRef\]](#)
47. Iwahori, Y.; Takada, Y.; Shiina, T.; Adachi, Y.; Bhuyan, M.K.; Kijirikul, B. Defect classification of electronic board using dense SIFT and CNN. *Procedia Comput. Sci.* **2018**, *126*, 1673–1682. [\[CrossRef\]](#)
48. U. Robots. Technical Specifications UR5. 2014. Available online: <https://www.universal-robots.com> (accessed on 13 March 2017).
49. Montalbo, F.J.P. A computer-aided diagnosis of brain tumors using a fine-tuned YOLO-based model with transfer learning. *KSII Trans. Internet Inf. Syst.* **2020**, *14*, 4816–4834.
50. Wei, Y.; Liu, X. Dangerous goods detection based on transfer learning in X-ray images. *Neural Comput. Appl.* **2020**, *32*, 8711–8724. [\[CrossRef\]](#)

51. Raza, K.; Song, H. Fast and accurate fish detection design with improved YOLO-v3 model and transfer learning. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 7–16. [CrossRef]
52. Liu, Y.; Zhang, X.; Gao, Y.; Qu, T.; Shi, Y. Improved CNN method for crop pest identification based on transfer learning. *Comput. Intell. Neurosci.* **2022**, *2022*, 9709648. [CrossRef]
53. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics (Version 8.0.0). [Computer Software]. 2023. Available online: <https://github.com/ultralytics/ultralytics> (accessed on 1 April 2024).
54. Xie, H.; Wu, Z. A robust fabric defect detection method based on improved RefineDet. *Sensors* **2020**, *20*, 4260. [CrossRef]
55. Misra, D. Mish: A Self Regularized Non-Monotonic Activation Function. 2019. Available online: <https://www.bmvc2020-conference.com/assets/papers/0928.pdf> (accessed on 7 October 2023).

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.