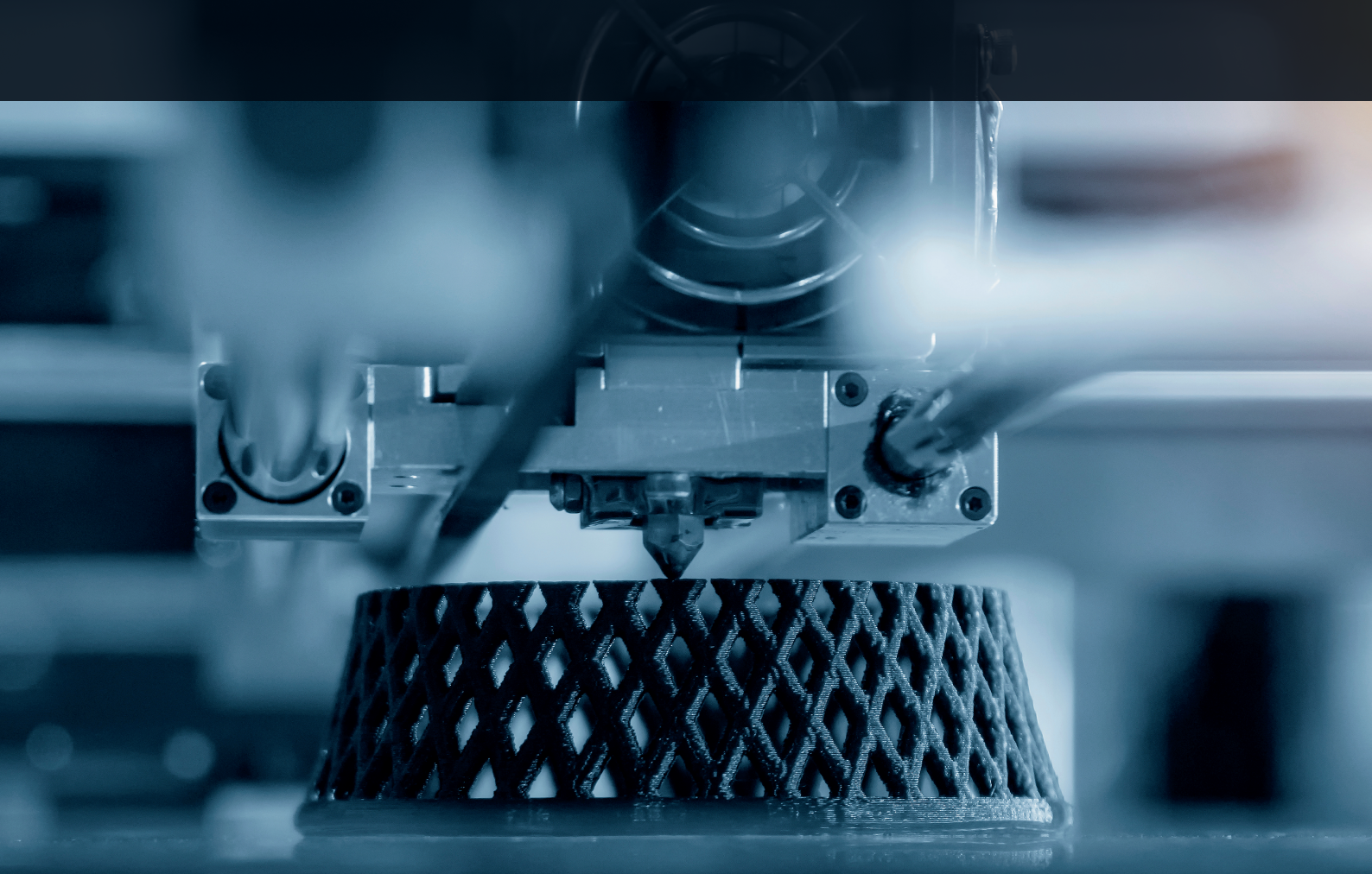
In-Situ Real-Time Fault Detection and Identification of FDM

Thesis - A.M.A. Tournoy



TU Delft

(this page is left intentionally blank)

In-Situ Real-Time Fault Detection and Identification of FDM

Thesis

A.M.A. Tournoy

In partial fulfilment of the requirements for the degree of

Master of Science

in Mechanical Engineering

at the Department Maritime and Transport Technology of the Faculty Mechanical Engineering

Chair:	Dr. J. Jovanova
Daily Supervisors	S. Kilambi, Dr. K. Masania , Z. Tocher
Institution:	Delft University of Technology
Place:	Faculty of Mechanical Engineering, Delft
Report Nr:	2024.MME.8952
Date:	February 9, 2025
Version:	1

In collaboration with Tanaruz Boats [1]

(this page is left intentionally blank)

Preface

This thesis represents the summit of my six year journey at Delft University of Technology. My time here has been truly unforgettable, allowing me to grow both personally and professionally while exploring my passions. Over the years, I have developed a strong interest in 3D printing, optimisation and automation. This thesis has only deepened that fascination. I eagerly look forward to the next chapter where I can apply and expand on everything I have learned. I hope that the findings presented in this work will contribute to the ongoing advancement of additive manufacturing processes and inspire further innovation in the field.

I would like to express my sincere gratitude to the supervisors who guided me through this project. Jovana's boundless enthusiasm consistently inspired and motivated me to push my limits. Kunal welcomed me into his research group, an extraordinary team of individuals whose academic excellence is truly inspiring. Shantha, my daily supervisor, was always approachable, patiently teaching me about convolutional neural networks and providing invaluable input throughout the process. Zander offered critical insights into large-scale printing, helping to elevate the impact of this project. His dedication and passion for the field remain a source of inspiration. Thank you for making this thesis possible.

To my friends in Delft, thank you for everything. In moments of success, you celebrated with me and during challenging times, you never failed to bring a smile back to my face. I am especially grateful to my solar team, whose drive and ambition taught me how to break boundaries, and to my futsal team for providing the most enjoyable thesis breaks. A special thanks goes to Carolina for her incredible help in sorting over 100,000 images—your support was invaluable. Another special thanks goes to Sanne, who helped me design the cover page and who always was up for study sessions with the most delightful coffee breaks.

To my dearest family, Iris, Tijs, Kurt, Geertrui and Ulf, the biggest thanks goes to you. Your belief in me, even when I failed to believe in myself, has meant everything. You were always there, lifting me up and creating a warm and loving home. Your support has been my foundation, and for that, I am forever grateful.

A.M.A. Tournoy

Delft, February 9, 2025

Abstract

Fused Deposition Modelling (FDM) is one of the most popular 3D printing technologies because of its affordability and accessibility. FDM, however, often suffers from printing errors that result in wasted time, materials and energy. To address these challenges, this thesis introduces a novel fault detection system for FDM printers. This system is designed to identify a broad range of errors without interrupting the printing process. To achieve a real-time detection system, an innovative multi-camera setup is designed, integrating two side cameras and one nozzle camera. Our hypothesis is that a system including three cameras can provide a more comprehensive view and can ensure more error types to be detected. Error detection is achieved using Convolutional Neural Networks (CNNs). This is a type of machine learning that excels at image recognition and pattern detection, making it well-suited for identifying printing errors in real-time models. Two CNN models are developed to classify images into common 3D printing errors: one model for the nozzle and another for the side cameras. The models were trained and validated on diverse datasets containing various shapes, infills, and augmented data. The nozzle camera model achieved a high validation accuracy of 97.68% with a low loss of 0.07464. The side camera model achieved comparable performance with a validation accuracy of 97.61% and validation loss of 0.1196. These two well performing models were for the first time ever integrated into a unified fault detection system based on a logic-driven priority framework. From this research, we learned that integrating multiple viewpoints into a logic-driven priority framework significantly improved the robustness of error classification, as many more error types could be detected in-situ and real-time. As a result, the integrated system successfully detected 12 common printing errors. In summary, this work shows the feasibility of developing a robust multi-input fault detection system to improve 3D printing. It paves the way for further research and implementation for complex integrated error detection and correction mechanisms.

Key Words:
fused deposition modelling, fault detection, convolutional neural networks, multi-camera integration

Contents

Preface i

Abstract ii

I Introduction 2

1 Background Information 3

2 Research Objectives 7

3 Thesis Outline 8

II Experimental Setup 10

4 Adaptations to the Printer 11

5 Camera Setup 13

5.1 Positioning of Cameras . . . 13

5.2 Selection of Cameras . . . 14

5.2.1 Nozzle Camera . . . 14

5.2.2 Side Cameras . . . 14

5.3 Camera Mounting . . . 14

6 Raspberry Pi System 16

III Data Acquisition 17

7 Print Settings and Error Simulation 18

7.1 Extrusion Errors . . . 19

7.1.1 Over Extrusion . . . 19

7.1.2 Under Extrusion . . . 19

7.2 Print Detail and Finish Issues . . . 19

7.2.1 Small Feature not Printed . . . 20

7.2.2 Foreign Body in Print . . . 20

7.2.3 Pillowing . . . 20

7.3 Support and Structural Issues . . . 20

7.3.1 Failing Support . . . 21

7.3.2 Poor Surface Quality at Support . . . 21

7.3.3 Weak Infill . . . 21

7.3.4 Infill Showing on Exterior of Print . . . 22

7.3.5 Poor Bridging . . . 22
7.4 Temperature and Cooling Issues . . . 23
7.4.1 Stringing . . . 23
7.4.2 Overheating . . . 23
7.4.3 Overhang Sagging . . . 23
7.5 Layer Alignment and Motion Issues . . . 24
7.5.1 Missing Layer . . . 24
7.5.2 Shifted Layer . . . 24
7.5.3 Ripples and Echoes . . . 25
7.5.4 Spaghetti Failure . . . 25
7.6 Layer Adhesion Issues . . . 25
7.6.1 First Layer Problems . . . 25
7.6.2 Warping . . . 26
7.6.3 Bed Separation . . . 26
7.6.4 Elephant Foot . . . 26
7.6.5 Delamination . . . 27

8 Dataset 28
8.1 Shapes and Infills . . . 28
8.2 Data Split . . . 30
8.3 External Validation Set . . . 31

9 Test Plan 32

10 Post-Processing and Data Augmentation 33
10.1 Post-Processing . . . 33
10.2 Data Augmentation . . . 34

IV Convolutional Neural Network Models 35

11 Introduction to Convolutional Neural Networks 36
11.1 Neural Networks . . . 36
11.2 Convolutional Neural Networks . . . 38
11.3 YOLOv8 Classification Model . . . 40

12 Model Training, Parameter Tuning and Transfer Learning 42
12.1 Training Process . . . 42
12.2 Parameter Tuning and Regularisation . . . 43
12.3 Transfer Learning . . . 44

13 Assessment of CNN Models 46
13.1 Validation and Test Data . . . 46
13.2 Assessment Metrics . . . 46
13.2.1 Accuracy . . . 46

13.2.2 Loss . . . 47
13.2.3 Confusion Matrix . . . 48
13.2.4 Precision, Recall and F1 . . . 49

V Results and Analysis **50**

14 Printing Results **51**

15 Performance of the CNN Models **53**
15.1 Nozzle Camera Model . . . 53
15.1.1 Parameter Tuning . . . 53
15.1.2 Training Graphs . . . 53
15.1.3 Accuracy and Loss . . . 54
15.1.4 Confusion Matrix . . . 54
15.1.5 Precision, Recall and F1 Score . . . 55
15.1.6 External Validation Set . . . 56
15.2 Bed Side Camera Model . . . 57
15.2.1 Parameter Tuning . . . 57
15.2.2 Training Graphs . . . 57
15.2.3 Accuracy and Loss . . . 57
15.2.4 Confusion Matrix . . . 57
15.2.5 Precision, Recall and F1 Score . . . 58
15.2.6 External Validation Set . . . 58

16 Discussion of the Models' Performance **60**
16.1 Interpretation of the Results . . . 60
16.1.1 Nozzle Camera Model . . . 60
16.1.2 Side Camera Model . . . 61
16.2 Comparison to Literature . . . 62
16.3 Strengths and Weaknesses of the Models . . . 64
16.3.1 Nozzle Camera Model . . . 64
16.3.2 Side Camera Model . . . 64

VI Fault Detection System **65**

17 Outline of the Detection System **66**

18 Logic of the Detection System **68**
18.1 Returning Individual Answers per Camera . . . 68
18.1.1 Occurence of Good, Unclear or Not Printing . . . 68
18.1.2 Consecutive Answers . . . 68
18.2 Integration of Camera Outputs . . . 70

19 Strengths and Weaknesses of the Fault Detection System **72**
19.1 Strengths of the Fault Detection System . . . 72
19.2 Weaknesses of the Fault Detection System . . . 72

VII Exploring Future Applications, Opportunities, and Their Impact **73**

20 Possibilities for Correction System **74**
20.1 Future Steps to Achieving a Correction System . . . 74
20.2 Correction per Error Type . . . 75
20.2.1 Extrusion Errors: Over Extrusion and Under Extrusion . . . 75
20.2.2 Finish Issues: Pillowing and Foreign Bodies . . . 75
20.2.3 Support and Structural Issues: Weak Infill and Poor Bridging . . . 76
20.2.4 Temperature Issues: Stringing and Overhang Sagging . . . 76
20.2.5 Layer Alignment Issues: Shifted Layers and Spaghetti Failure . . . 76
20.2.6 Layer Adhesion Issues: Warping and Delamination . . . 76

21 45° Belt Printing **78**
21.1 Relevance for 45° Printing . . . 78
21.2 Adjusting System to 45° Printers . . . 79

22 Large-scale 3D printing **80**
22.1 Relevance for Large-Scale Printing . . . 81
22.2 Adjusting System to Large-Scale Printers . . . 82

23 Possible Impacts of the Fault Detection System **83**
23.1 Desktop Printing . . . 83
23.2 Large-Scale Printing . . . 83

24 Significance of This Work **85**
24.1 Key Contributions . . . 85
24.2 Insights Gained . . . 85
24.3 Broader Impact and Future Directions . . . 86

VIII Conclusion **87**

25 Conclusion **88**

26 Future Recommendations **90**

References **91**

A Confusion Matrices **95**
A.1 Nozzle Camera Model . . . 95
A.2 Side Cameras Model . . . 98

B Precision, Recall and F1 Scores **101**

B.1 Nozzle Camera Model 101

B.2 Side Cameras Model 102

C Research Paper **103**

PART I

Introduction

1| Background Information

Additive manufacturing, commonly known as 3D printing, is an innovative technology that rapidly and precisely transforms computer-aided design (CAD) files into physical structures [2, 3]. These structures are created by systematically joining materials layer-by-layer, using substances such as resins, inks and powders [4]. By adding material to form structures, custom design and complex structures can be fabricated while significantly reducing material waste compared to conventional manufacturing methods [5]. Based on their working principles, additive manufacturing techniques are classified into the following categories, as illustrated in Figure 1.1.

Extrusion-based printing techniques involve extruding molten or semi-molten material through a nozzle to build structures layer by layer. This layered approach is used in techniques such as fused deposition modelling (FDM) and direct ink writing (DIW). FDM is widely used for its compatibility with thermoplastics, while DIW employs viscous or shear-thinning inks [4–7]. Beyond these methods, less-commonly used printers utilise granulate as in-feed material. Fused Granular Fabrication (FGF) has become a promising technique for additive manufacturing, reducing melt solidification steps and eliminating the filament extruder system. This simplification promotes a greener approach to distributed recycling, making it a viable option for desktop 3D printing and significantly lowering material processing costs [8].

Powder Bed Fusion (PBF) represents another category of additive manufacturing techniques. In this contactless technique, powder materials are selectively fused using a heat source such as a laser or electron beam. Examples include selective laser sintering (SLS), selective laser melting (SLM) and electron beam melting (EBM). These processes are particularly notable for fabricating complex geometries and mechanical metamaterials without support structures [4, 9, 10].

Vat Photopolymerisation, on the other hand, relies on light to cure photopolymer resin layer by layer. High-resolution and smooth-surface structures are created using techniques such as stereolithography (SLA) and digital light processing (DLP). While SLA employs a focused laser to achieve this, DLP uses a projector for rapid production, making it a faster alternative for certain applications [4, 5, 11–13].

Direct Energy Deposition (DED) takes a different approach by using a movable nozzle to melt and deposit material, such as powder or wire, onto a substrate. The precise material placement makes this method particularly suitable for repairing large metal parts or creating complex components. One notable example of DED includes Laser Engineered Net Shaping (LENS) [10, 14–16].

Finally, Inkjet Printing operates similarly to conventional 2D printing. These inkjet methods deposit droplets of material onto a bed, which are cured using UV light. This principle is used in PolyJet printing. PolyJet is a methodology that enables multi-material printing with excellent resolution and surface finish. In contrast, binder jetting uses a polymeric binder to bond powders, such as metals and ceramics. This method offers a unique approach to creating durable and intricate structures [4, 11, 17, 18].

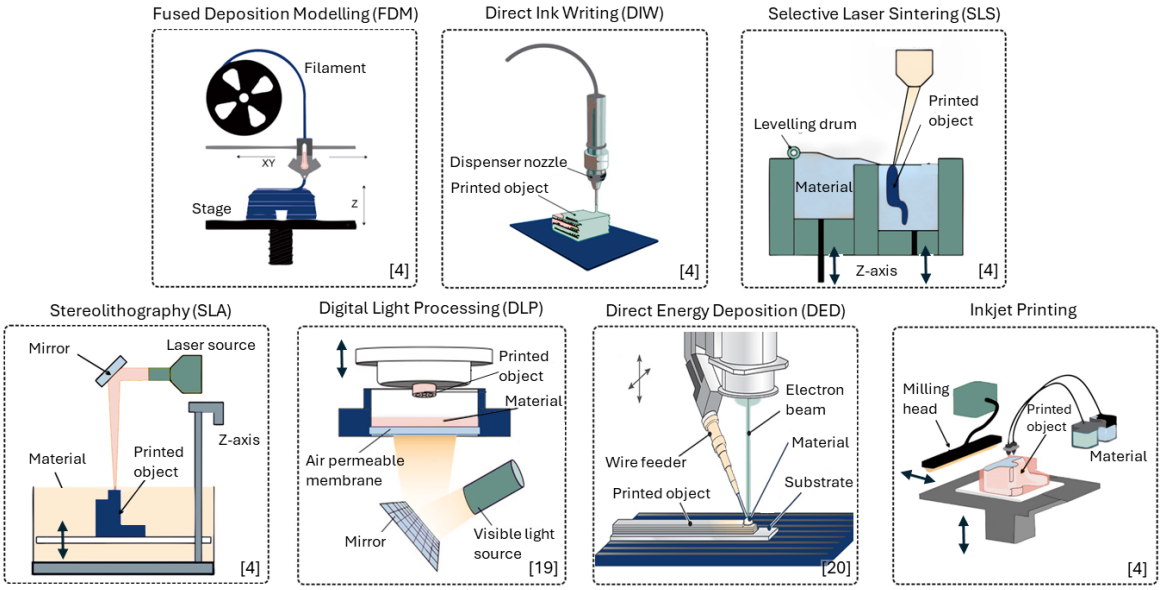


Figure 1.1: Additive manufacturing techniques: FDM, DIW, SLS, SLA, DLP, DED and inkjet printing (adapted from [4, 19, 20])

Of all aforementioned methods, FDM is the most commonly used additive manufacturing technique due to its compatibility with widely available materials and ease of use [4–6]. In FDM, a continuous thermoplastic filament is unwound and liquefied to a semi-liquid state using high temperatures in the nozzle. Then, the material gets extruded under pressure [5, 9, 11, 21].The extruded filament deposits lines onto previously made layers, which solidify upon cooling to form bottom-up 3D structures [13, 22, 23]. This layer-by-layer approach offers several advantages, including cost-effectiveness, user-friendliness and minimal post-processing requirements. However, drawbacks include relatively low resolution (100–200 [µm]) [17], low surface quality and a high frequency of defects [4, 24].

Defects in printed components occur frequently and limit the potential and benefits of FDM manufacturing [2, 25–27]. These FDM defects are particularly prevalent among inexperienced users, who encounter a failure rate of around 20% [28–31]. This high failure rate can be attributed to the open-loop nature of 3D printing. To address this, a closed-loop real-time control for FDM printers is required.

Several methodologies have been studied to detect and correct common mistakes in FDM. These methodologies consist of analysing image data [2, 26–28, 30–36], temperature [29, 37], acoustic emissions [38–41] and electrical resistance [25]. Among these, image data analysis has proven to be particularly effective, with studies either using a single or multi-camera setup [30, 42]. These camera-based approaches are adopted in various studies. An overview of the methodologies of these studies is presented in Figure 1.2. For example, Jin et al. [33, 34] employed deep learning with Convolutional Neural Networks (CNN) to detect and correct under extrusion and over extrusion errors, shown in Figure 1.2A. Similarly, studies focused on using CNN models to identify and limit interlayer imperfections such as warping and delamination [32, 35]. The work of Brion et al. [35] is depicted in Figure 1.2B. Brion and Pattinson [26] employs image classification to categorise four printing parameters, i.e. flow rate, lateral speed, z-offset, and hotend temperature, into low, good, and high. This classification is then utilised for in-situ correction of these parameters. The methodology of this study is presented in

Figure 1.2C. In another study, Petsiuk and Pearce [28] explored the use of a single-camera system to project top and side views for height validation, trajectory matching and texture analysis, depicted in Figure 1.2D. Additionally, Petsiuk and Pearce [30], Nuchitprasitchai et al. [31] used histogram differences and image comparison techniques to identify defects by comparing real-time images to 2D or 3D render references.

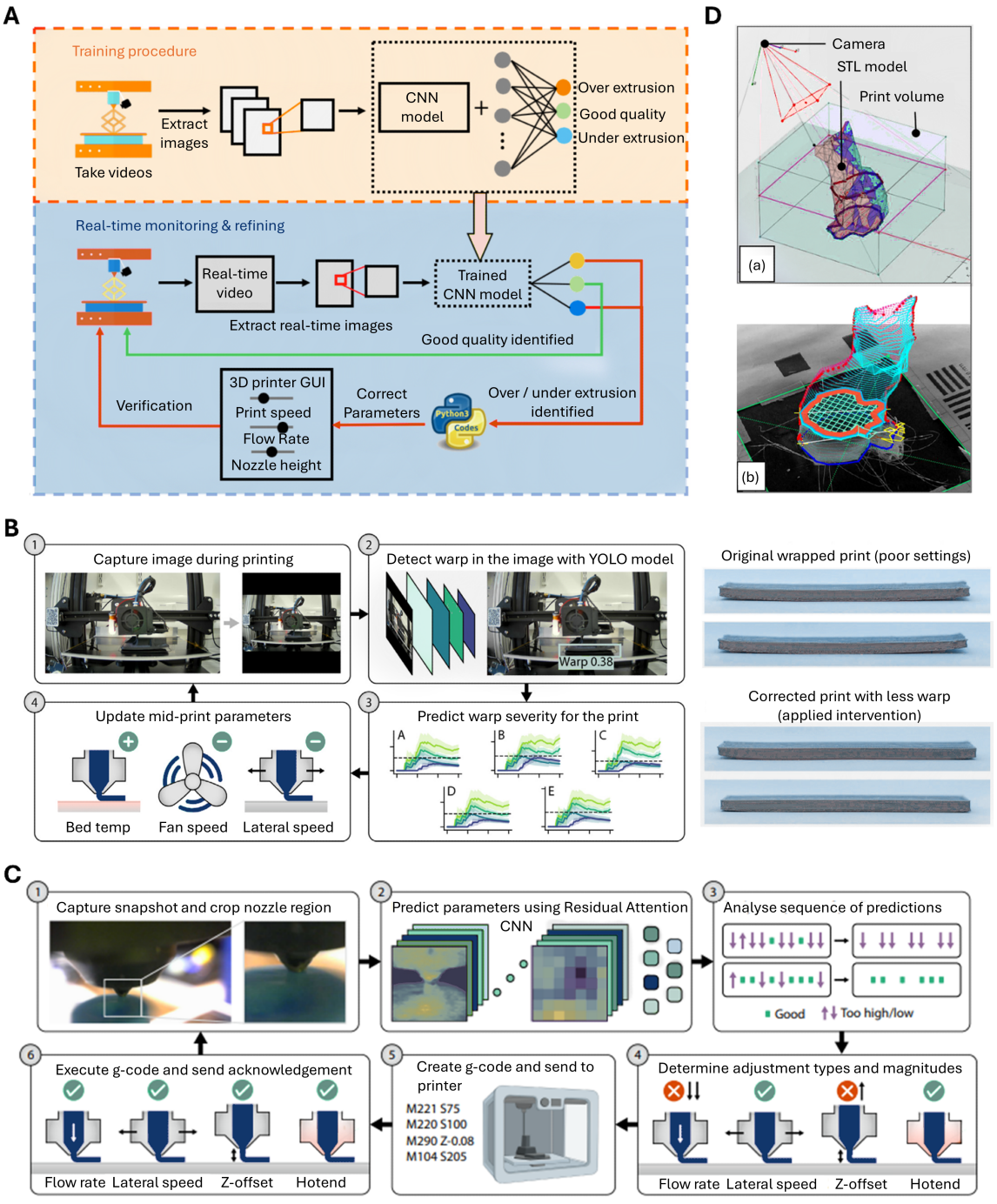


Figure 1.2: Existing fault detection and correction methodologies. A: Jin et al. [34] introduced a extrusion error detection and correction system. B: Brion et al. [35] designed a warping detection and correction system. C: Brion and Pattinson [26] created a detection and correction of print parameters. D: Petsiuk and Pearce [28] made a error detection system based on height validation, trajectory and texture analysis of the printed part compared to a referenc (a) the camera position relative to the STL model, (b) G-Code trajectories projected.

Despite these advancements, several gaps remain in the current methodologies. First, many studies only focus on specific defect types, such as solely focussing on delamination, warping, or extrusion errors [32, 33] and without offering integrated feedback systems to detect and correct multiple types of errors in real-time. This narrow focus limits the overall effectiveness of these systems. Secondly, for many error types, there is still no dedicated detection system available. Figure 1.3 depicts an overview of common mistakes in FDM, marked yellow if a detection system exists and green if a complete feedback system capable of correcting these errors exists. Thirdly, many existing detection systems are not in-situ, causing delays in the printing process. For example, the work by Petsiuk and Pearce [28] covers a comprehensive range of error types but is computationally inefficient, adding approximately 50% extra time to the printing process and lacking a correction procedure. Nuchitprasitchai et al. [31] faced limitations due to the time required for analysis, restricting real-time monitoring capabilities. These delays are particularly evident in methods that rely on calculating histogram differences between measured and synthetic reference images. This process is too time-consuming and unsuitable for complex geometric shapes or varying lighting conditions [30, 31]. However, using a CNN, real-time detection of errors becomes feasible, addressing the time-consuming and computational inefficiencies of traditional methods. Hence the most promising techniques involve these CNNs. Additionally, static camera setups often miss errors occurring in different parts of the print or under varying lighting conditions [35]. Single-camera systems often lack the ability to capture errors on the backside of the print or during rapid printing processes. Therefore, literature suggests that current approaches should be expanded to incorporate multi-camera setups and possibly integrate other sensor data for more comprehensive in-situ error detection and correction [26, 27, 32, 33]. The main goal of this thesis is to cover these gaps by designing a multi-camera detection system utilising integrated CNN models capable of detecting various common 3D printing errors, for a small desktop printer.

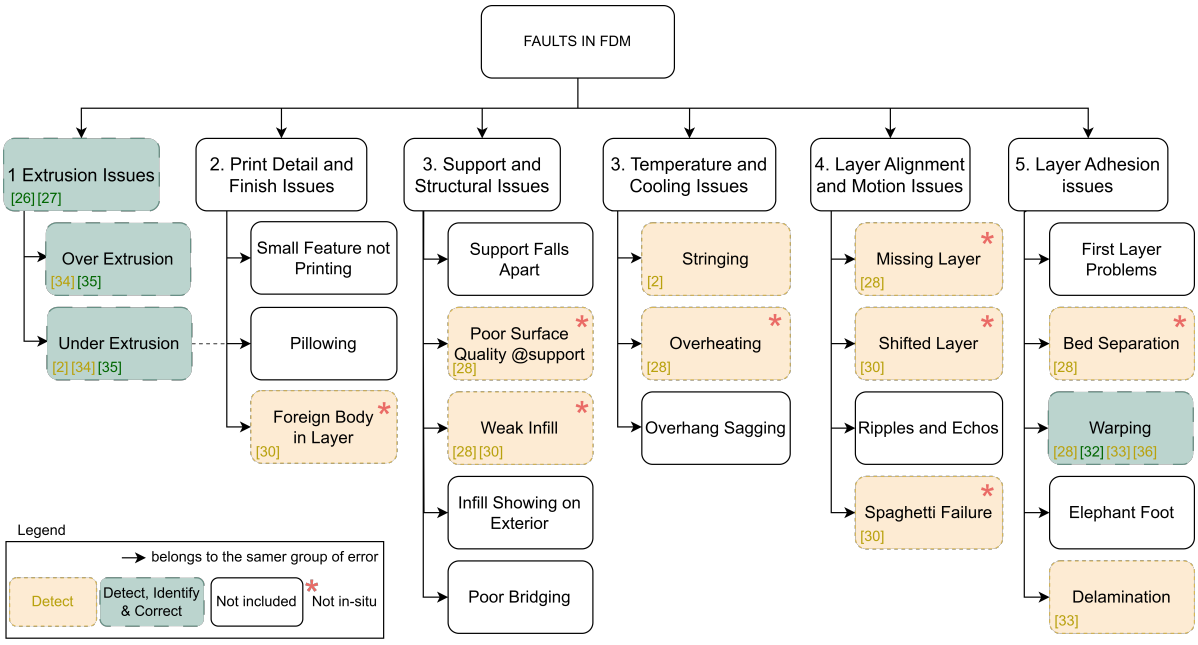


Figure 1.3: Common errors in FDM, and whether there already exists systems to detect and correct these [2, 26–28, 30, 32–36]

2| Research Objectives

This thesis aims to develop an integrated fault detection system for FDM printers utilising a multi-camera setup and convolutional neural networks (CNN) models. The primary goal is to design a rapid and highly accurate detection system capable of identifying a wide range of common printing errors real-time and in-situ. This is achieved using image classification on a small-scale desktop printers as a foundational approach. The approach aims to lay the groundwork for future developments, including the integration of object detection methods and corrective mechanisms.

To guide this research, the following main question and subquestions are posed. This systematically addresses various aspects of the FDM fault detection system.

Main Research Question:
"Can the performance of FDM printers be enhanced by developing an integrated fault detection system, based on a multi-camera setup, using convolutional neural network models?"

Subquestions:

1. How does a multi-camera system compare to single-camera and sensor-based systems in terms of accuracy and amount of errors covered?
   This question evaluates the effectiveness of the proposed multi-camera system. It compares model accuracies and the amount of error classes covered to that of single-camera systems reported in the literature. It aims to assess whether the additional complexity introduced by multiple cameras yields significant improvements in detecting and identifying faults.
2. Which logic framework can integrate data from multiple cameras into a detection system?
   This question explores creating a cohesive logic framework to combine data from multiple cameras effectively. This involves determining which cameras are best suited to detect specific errors. When these errors are detectable by multiple cameras, the system must integrate the perspectives to make a unified decision. This involves the algorithms and decision-making processes necessary to optimise the detection mechanism.
3. What are the future steps for integrating a correction mechanism into the fault detection system?
   Beyond detecting faults, incorporating a correction mechanism is essential for creating a robust and autonomous system. This question examines potential approaches to integrate such mechanisms. This would enable real-time error correction that can be employed in future research.
4. What additional steps are required to apply this method to large-scale FDM robots?
   To broaden the system's applicability, this question explores how the methodology can be scaled for industrial large-scale use. This involves addressing challenges unique to large-scale FDM robots, such as increased build volumes, higher environmental variability and the need for additional computational resources, while preserving the proposed system's accuracy and speed.

3| Thesis Outline

This thesis report contains the following structure. Part II describes the experimental setup, including the technical modifications made to the 3D printer. These changes enable effective fault detection during operation. Adaptations to integrate the multi-camera setup are thoroughly described. This includes the rationale behind camera positioning, the selection criteria for suitable cameras and mounting mechanisms used to ensure stability and image quality. Additionally, the role of the Raspberry Pi system is explained, highlighting its function in managing image acquisition and facilitating remote control. Together, these steps establish a solid foundation for implementing the detection system.

Part III focuses on data acquisition. This includes the methodology for generating and preparing the dataset used to train and evaluate the detection system. It begins with an overview of the printing settings and controlled simulations of errors designed to replicate real-world conditions. Then, the structuring of the dataset, including class balancing and file organisation, is described in detail. A systematic test plan is developed to ensure comprehensive coverage of potential printing errors. Furthermore, techniques for post-processing and data augmentation are explained, aiming to enhance dataset diversity and robustness to improve model performance.

Part IV examines the deep learning model employed for high quality fault detection. An introduction to the relevant deep learning techniques provides context for the architecture and capabilities of the chosen model. The training process, including the use of transfer learning to leverage pre-trained models, is outlined. Key considerations during training, such as hyperparameter tuning and the application of regularisation techniques, are discussed. The part concludes with describing diverse ways to evaluate the performance of CNN models.

Part V presents the results derived from the experiments conducted during this research. This includes the results of the printing tests and the results of the trained CNN model. The effectiveness of the CNN models in detecting faults is evaluated, emphasising their accuracies and losses, using the internal and external dataset. A critical appraisal of the results is included, addressing both the successes and the limitations observed. This part aims to answer the first subquestion posed in Chapter 2.

Part VI explores the fault detection system, detailing how the CNN models are integrated into a real-time detection framework. This logic framework of the detection system is explained, with a step-by-step description of how data is collected, processed and used to identify faults during operation. The practical considerations involved in implementing the system in a production environment are discussed. This part addresses the second subquestion.

Part VII extends the scope of this research by exploring the incorporation of a correction mechanism. Furthermore, the scope is extended by considering its applicability to large-scale 3D printing and 45° printing. The challenges and opportunities associated with adapting the fault detection system for industrial-scale printers are explored. This part highlights how the findings of this thesis can contribute to broader advancements in additive manufacturing. This part answers the third and fourth research subquestions. Lastly, this part covers the potential impact the fault detection offers, considering the amount of materials and costs that would be saved when this system is deployed.

Part VIII concludes the thesis, reflecting on the initial research objectives and evaluating the extent to which they have been achieved. Key findings from the experimental and analytical work are consolidated to provide a clear overview of the contributions made by this study. Future recommendations are proposed, focusing on improving detection accuracy, integrating correction mechanisms and extending the applicability of the system to other manufacturing contexts.

PART II

Experimental Setup

This part outlines the experimental setup used to design and evaluate the fault detection system, featuring a modified Prusa Mk3s printer with glass fibre-reinforced polypropylene (PP-GF30) filament. To enable real-time fault detection, a new multi-camera setup is developed, with two Raspberry Pis configured to handle image processing and monitoring tasks.

The goal of this section is to provide a detailed description of the experimental setup used throughout the thesis. Chapter 4 describes the modifications made to the Prusa Mk3s, including changes to the spool holder, print bed and nozzle. Chapter 5 explains the selection, placement and mounting of cameras used for monitoring the printing process. Finally, Chapter 6 covers the configuration of the Raspberry Pis to process images for fault detection and to integrate the setup with the 3D printer for real-time monitoring and correction.

4| Adaptations to the Printer

For this research, a Prusa Mk3s FDM printer [43] was selected and adapted to print glass fibre-reinforced polypropylene filament. The glass fibre-reinforced polypropylene filament (PP-GF30) from TransMare [44] was chosen, because PP-GF30 is commonly used in industrial large-scale 3D printing robots that fabricate complex structures, including leisure boats [1]. Given its uncommon use in desktop 3D printing, adaptations were required to ensure consistent quality and optimal printer performance.

The initial step involved feeding the material into the printer and conducting temperature tests to determine the ideal printing parameters. Different tower tests involved printing at different bed temperatures and different fan speeds. Each section of the tower was printed at a progressively higher nozzle temperature, allowing for visual inspection of the optimal settings, as shown in Figure 4.1. The optimal printing parameters are summarised in Table 4.1.

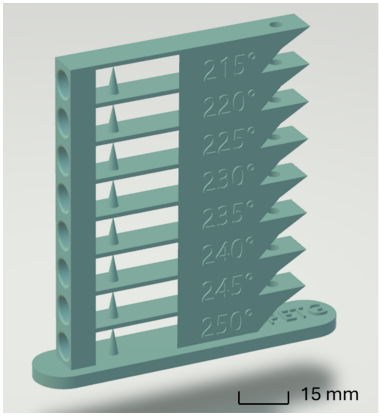


Figure 4.1: CAD file of the temperature tower test

Parameter	Value
Nozzle temperature	235 °C
Nozzle temperature first layer	240 °C
Bed temperature	40 °C
Maximum fan speed	40 %
Bridges fan speed	70 %
Disable fan for the first	5 layers

Table 4.1: Printing parameters

The material was supplied on a large spool, weighing 4.1 kg, which was too large for the printer's standard spool holder. To resolve this, a custom variable spool holder was printed, along with a filament guide to prevent sharp bends that could damage the material. Additionally, due to the abrasive nature of glass fibres, the standard brass nozzle was replaced with a steel nozzle to withstand wear over extended use. A nozzle diameter of 0.4 [mm] was chosen, which is a commonly used size for general-purpose printing [45]. Furthermore, the filament's dark colour required painting the print bed white with temperature-resistant paint to improve the visual contrast between the printed part and the bed. The bed itself was coated with polypropylene tape to prevent warping, a common issue with polypropylene, since the material adheres poorly to surfaces other than polypropylene. These modifications are summarised in Figure 4.2.

One standard test model, commonly referred to as the Benchy [46], was used to verify the effectiveness of the chosen printing parameters and implemented adaptations. The Benchy model is specifically designed to challenge 3D printers with its complex geometrical features. It tests critical factors such as overhang quality, bridging performance and dimensional accuracy. This test offers a quick and efficient

method to assess the printer's performance under the defined settings. As depicted in Figure 4.3, the results of this test demonstrated high print quality, confirming the suitability of the chosen parameters and adaptations.

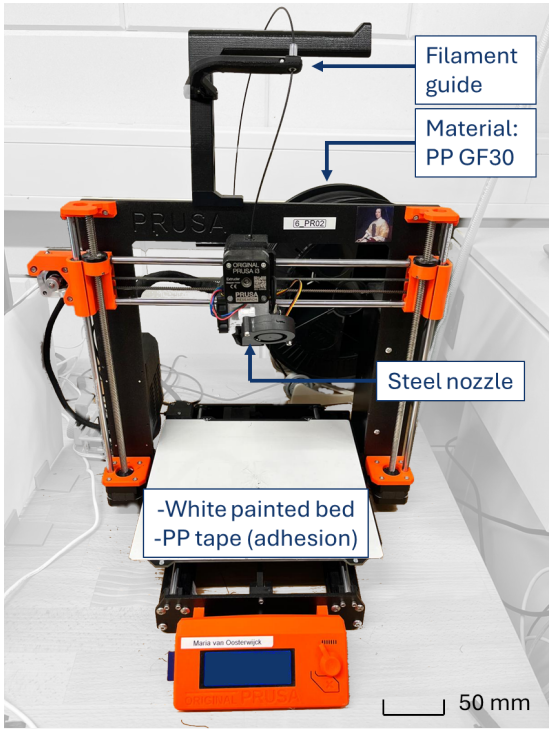


Figure 4.2: Adaptations made to the printer: the material is PP GF30, added filament guide, a steel nozzle and the bed is painted white and a layer of PP tape is added for adhesion



Figure 4.3: Result of the successful Benchy print

5| Camera Setup

5.1. Positioning of Cameras

To effectively capture common 3D printing errors, careful positioning and selection of cameras are of utmost importance. Errors related to extrusion require a nozzle camera, which is mounted close to the nozzle. A camera capable of capturing fine details at close range was selected for this purpose. To detect other errors visible from the sides of the print, two cameras are mounted on the print bed to ensure visibility for three out of four sides. For the bed mounted cameras, multiple setup configurations are considered: the cameras can be mounted on the Gantry of the printer, in front and back of the printer, on the corners either diagonally across each other or on the same side as graphically illustrated in Figure 5.1. Each configuration has benefits and limitations. In all four setups, only the outer shell of the print is captured. Moreover, only the sides at which the cameras are positioned can be captured but strategic positioning of the print can maximise coverage.

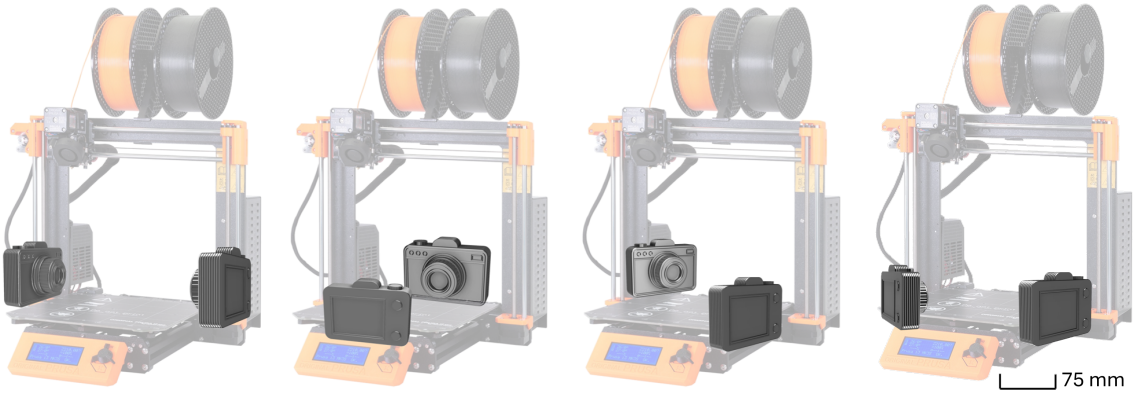


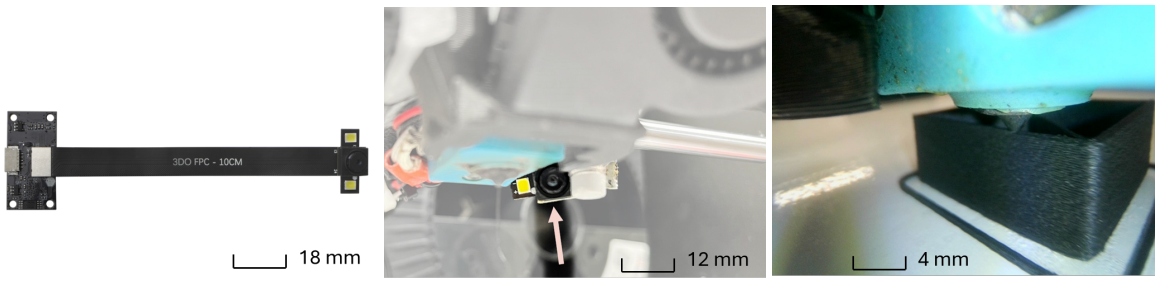
Figure 5.1: Different positions for the two bed mounted cameras: on the gantry, in front and back of the printer, diagonally in the front and back and two cameras in front (adapted from [47, 48])

The first option, mounting the cameras on the gantry, avoids collisions with moving parts and is supported by existing open-source brackets. However, because the bed moves while the gantry is fixed, the object shifts relative to the camera. This setup requires therefore either a wide field of view, achieved with fisheye lenses or increased camera distance or a system to capture images at specific bed positions. These solutions increase complexity and cost, making this option impractical. The second option, with cameras placed at the front and back of the printer, provides a narrower field of view compared to diagonal placements and was therefore not preferred. The third and fourth options involve diagonal and same-side corner placements, respectively. The diagonal setup offers better coverage of the print, as both cameras can view more surfaces. However, one camera must be installed near the printer's power supply, which adds complexity to set up. For larger-scale printers, the same-side corner placement (option 4) appears to be more feasible, as it avoids interference with other components and simplifies installation. Because of a trade-off between visibility, complexity to install and the applicability to large-scale printing, the fourth option was chosen for small-scale setups.

5.2. Selection of Cameras

5.2.1. Nozzle Camera

The 3DO Nozzle Camera Kit [49] was chosen to provide close-up images of the deposited material. This camera, designed specifically for high-temperature environments, resulted in the desired image quality. Furthermore, the inclusion of a silicone cover over the heater block effectively minimised reflections. Figure 5.2 illustrate the camera, its position and a sample image.



(a) 3DO nozzle kit [49] **(b)** Positioning of the nozzle camera **(c)** Image from nozzle camera

Figure 5.2: 3DO nozzle camera

5.2.2. Side Cameras

For side monitoring, the Raspberry Pi HQ Camera (CS Mount) [50], equipped with a 6mm 3MP-HD wide-angle lens [51], was selected. This configuration provides a wide field of view and manual focus capability. This ensures a complete coverage of printed parts within a field measuring at least 80×80×80 [mm]. Figure 5.3a illustrates the field of view of the bed from the side cameras. Figure 5.3b shows the camera itself and Figure 5.3c depicts a sample image.



(a) Dimensions of the field of view of the printing bed (top view) in [mm] **(b)** Raspberry Pi HQ camera (adapted from [50, 52]) **(c)** Image from Raspberry Pi camera

Figure 5.3: Side camera: Raspberry Pi HQ

5.3. Camera Mounting

The 3DO Nozzle Camera was mounted using a custom-designed bracket attached to the hotend of the 3D printer. The bracket is secured with the screw located behind the auto bed levelling sensor. This bracket is shown in Figure 5.4 highlighted in pink, showing mounting of the nozzle camera using three different magnification levels. The custom nozzle camera bracket was fabricated from PP-GF30 to withstand the high temperatures near the nozzle.

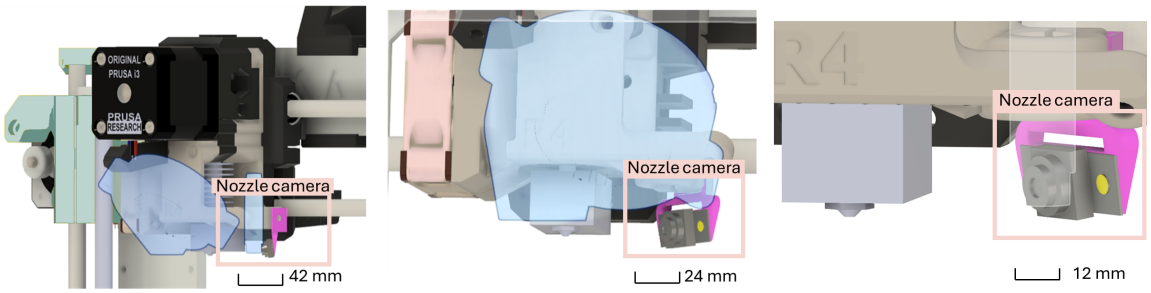


Figure 5.4: Mounting of the nozzle camera using three levels of magnification, for better visibility the fan and levelling sensor components are either hidden or rendered transparent

Mounting of the bed cameras is based on an existing bed camera bracket from Printables [53] that has been adapted for this application. The bracket assembly per camera consists of four interconnected components: a bed bracket, a swing arm, a cradle and a case for the camera. The bed bracket slides onto the bed, ensuring secure attachment. The swing arm provides the ability to rotate the camera left and right, while the cradle enables the camera to be angled up or down. The case holds the camera securely in position. To withstand the elevated temperatures from the heated bed, the bed brackets and swing arms are fabricated from PP-GF30, as components made from the commonly used material polyactic acid (PLA) were observed to deform under such conditions. In contrast, the cradles and cases are made from PLA. Figure 5.5 illustrates the complete brackets, with the cradles and camera holders in white PLA and the bed brackets and swing arms in black PP-GF30.

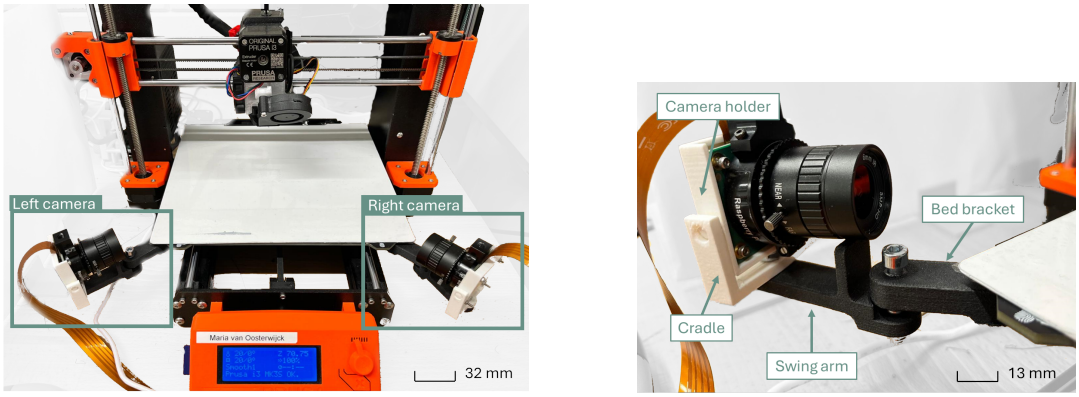


Figure 5.5: Mounting of the left and right bed cameras, using four parts: a bed bracket, swing arm, cradle and camera holder

6| Raspberry Pi System

The setup consists of two Raspberry Pis to collect the necessary images, which is graphically visualised in Figure 6.1. One Raspberry Pi 3 [54] is connected to the 3D printer, allowing to run OctoPrint [55]. Octoprint is open-source software enabling remote control of the 3D printer. OctoPrint also facilitates the extraction of real-time printer data for further analysis, such as the nozzle temperature, the bed temperature, the Z-offset etc. The other Raspberry Pi 5 [56] is connected to the two bed cameras and the nozzle camera. This configuration allows the cameras to capture images and save them directly to a hard drive during operation. A custom script has been written to automate the image capture process and simultaneously log the printing parameter data extracted from OctoPrint, ensuring all necessary data is appropriately stored. However, the data extraction proved to be too computationally intensive and was therefore disabled during data collection.

During the data collection process, a power issue was encountered with the nozzle camera. The Raspberry Pi 5 was unable to provide sufficient power to the 3DO nozzle camera, rendering it unrecognisable. Consequently, the nozzle camera was connected to a laptop, which supplied adequate power for data collection. This workaround, however, caused a loss of synchronisation between the nozzle camera images and the side camera and the printer data. Despite this, synchronisation was not critical for the collection of training data, as the primary objective was to gather labelled image data rather than time-sensitive correlations.

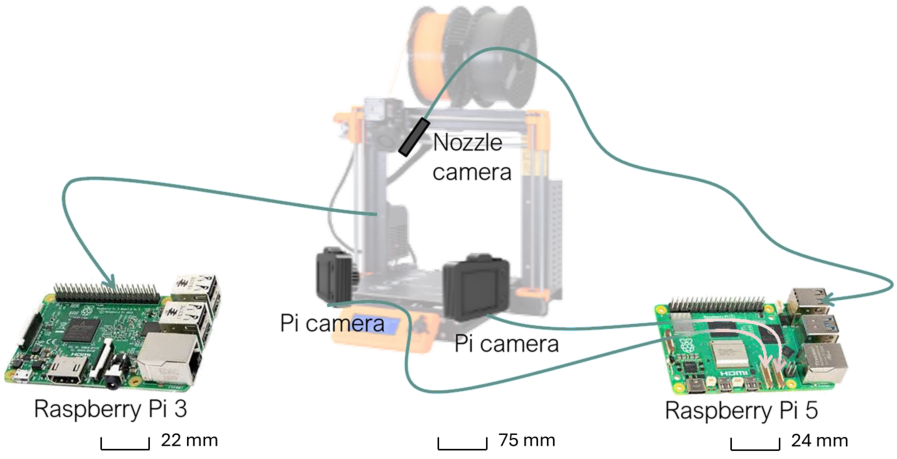


Figure 6.1: Graphical setup, consisting of three cameras and two raspberry pi units (adapted from [47, 48, 57, 58])

PART III

Data Acquisition

To develop robust neural network models capable of detecting common 3D printing errors, a substantial and diverse dataset is required. This dataset must include images of both successful prints and prints exhibiting a wide range of errors. To gather this data, various tests were designed and executed, capturing images under different conditions and print settings.

To ensure data diversity, a variety of shapes and infill patterns were selected. After collecting raw data, augmentation techniques were applied to further increase the size and variety of the dataset. This approach optimised the time spent on testing while ensuring comprehensive data coverage. The tests were carefully crafted to minimise duration while maximising data quality. Data gathering followed a structured plan that purposefully triggered common 3D printing errors by manipulating specific parameters. The print parameters were deliberately adjusted based on reverse-engineering solutions commonly used to correct certain errors. For example, over extrusion is typically addressed by lowering the extrusion rate; thus, over extrusion was induced by intentionally increasing the extrusion rate.

In Chapter 7, the print parameters and configurations used to provoke various errors are discussed. Chapter 8 outlines the data set design, covering shapes and infill patterns selected to ensure data variety. Chapter 9 describes the preparations required for testing, as well as the execution of the tests themselves. Chapter 10 covers post-processing steps and data augmentation techniques applied to the collected images.

7| Print Settings and Error Simulation

The development of a fault detection system requires a comprehensive understanding of the types of errors that can occur during the FDM process. These errors can be categorised into groups, including good prints, extrusion issues, detail and finish errors, support and structural issues, temperature-related problems, layer alignment and motion issues and adhesion failures, as summarised in Figure 7.1 [59–61].

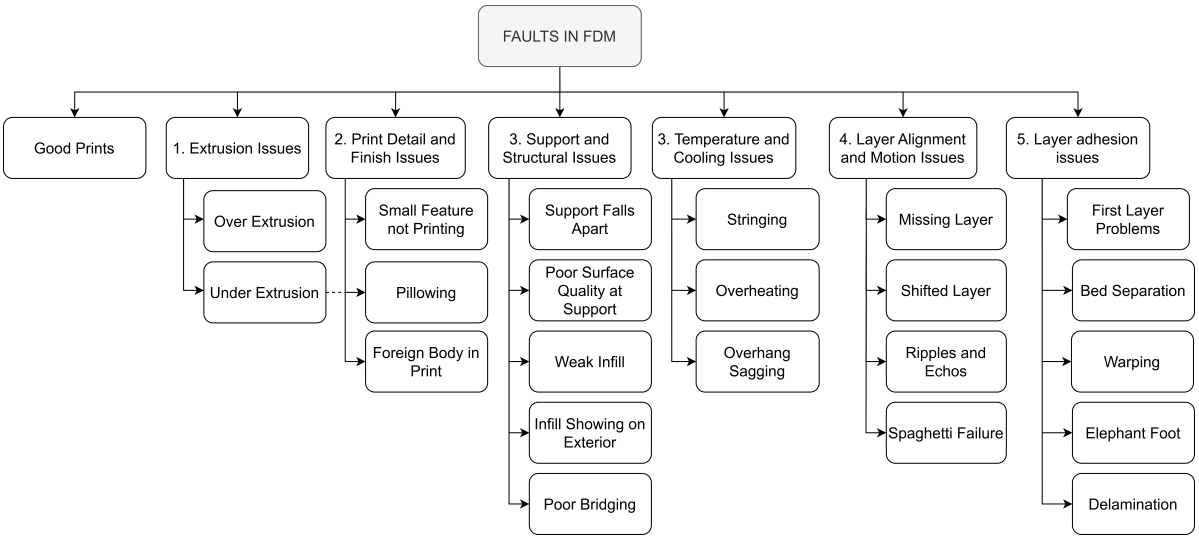
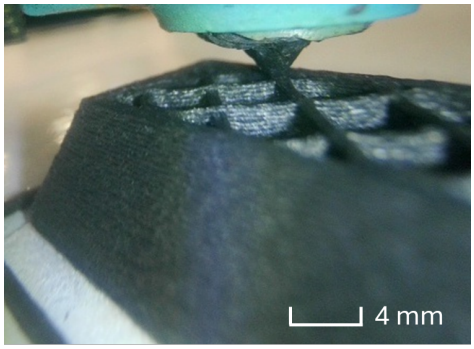
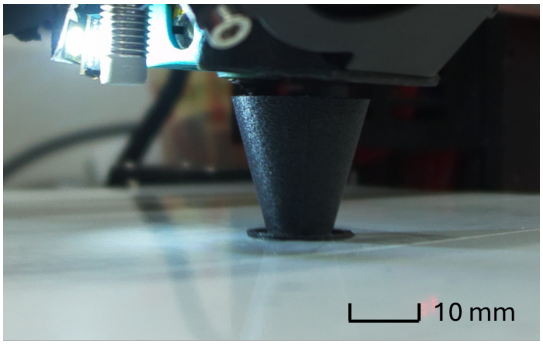


Figure 7.1: Common printing errors in FDM, sorted according to their nature

It is essential to have extensive datasets representing each error. The print settings of these datasets were generated through reverse-engineering optimal printing solutions into these faults, as documented in [59–61]. The configurations for inducing these errors are discussed in Section 7.1 through Section 7.6. The printing parameters were adjusted during slicing using PrusaSlicer [62]. This is a software commonly employed for converting 3D models into printer instructions, referred to as the G-code. Alternatively, printing parameters were manipulated directly during the printing process. For the dataset of good prints, standard parameters, as outlined in Chapter 4, were applied. Figure 7.2 depicts example images of good prints from both the nozzle camera and one of the side cameras.



(a) Good print, image taken using the nozzle camera



(b) Good print, image taken using the side camera

Figure 7.2: Example images of good prints

7.1. Extrusion Errors

Extrusion errors occur when the filament is either over or under extruded during the printing process, leading to defects such as excess material deposition or insufficient material flow. The nozzle camera is particularly effective in capturing these errors due to its detailed view of the filament deposition on the print bed.

7.1.1. Over Extrusion

Over extrusion is characterised by the excessive deposition of filament, resulting in rough surfaces, unwanted bulges or overlapping material layers, as shown in Figure 7.3. This issue is caused by incorrect settings: an increased material flow rate, a high extrusion multiplier or an incorrect nozzle diameter configuration. For instance, specifying a nozzle diameter higher than the true diameter can lead to over extrusion. The parameters used to induce over extrusion, together with their nominal value, to obtain good prints, are summarised in Table 7.1.

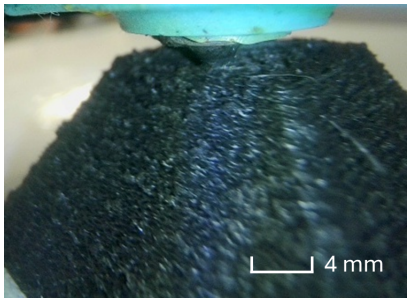


Figure 7.3: Example of over extrusion, showing rough surfaces and unwanted bulges

Test	Parameter	Value	Nominal
1	Flow rate	130 - 170%	100%
2	Flow rate	130 - 170%	100%
2	Nozzle diameter	0.8 [mm]	0.4 [mm]

Table 7.1: Printing parameters to induce over extrusion

7.1.2. Under Extrusion

Under extrusion occurs when insufficient filament is extruded, resulting in gaps, weak layers and inconsistent surfaces, as depicted in Figure 7.4. This defect often arises from low flow rate settings, an incorrect nozzle diameter or suboptimal temperature settings that reduce filament flow. The parameters used to achieve under extrusion are outlined in Table 7.2.

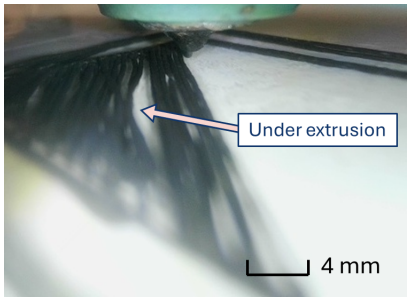


Figure 7.4: Example of under extrusion, recognisable by gaps

Test	Parameter	Value	Nominal
3	Flow rate	30 - 70%	100%
4	Flow rate	30 - 70%	100 %
4	Nozzle diameter	0.25 [mm]	0.4 [mm]

Table 7.2: Printing parameters to induce under extrusion

7.2. Print Detail and Finish Issues

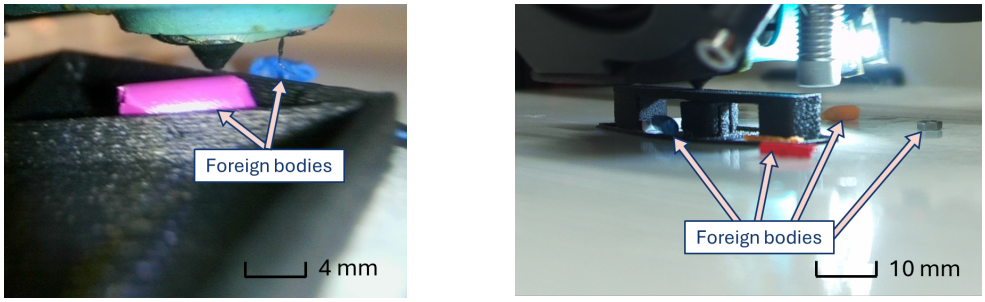
Print detail and finish issues include challenges such as small features not being printed, foreign bodies embedded in prints and pillowing. These errors impact the final appearance and quality of prints.

7.2.1. Small Feature not Printed

Small features may fail to print correctly if they are smaller than the resolution of the printer. Since identifying this issue from images alone is not possible without prior knowledge of the intended part, this fault is excluded from the scope of the detection system. Addressing such issues would require incorporating part-specific information into the analysis, which is beyond the current scope.

7.2.2. Foreign Body in Print

Foreign bodies embedded in the print layers often result from unclean working environments, where small particles, such as dust or debris, contaminate the filament or print bed. These defects are visible in images captured by both the nozzle and side cameras. To generate data for foreign body detection, small objects such as M2 nuts, tape residues and pieces of filament were manually placed within the printed parts, as shown in Figure 7.5.



(a) Foreign bodies in print, taken with the nozzle camera

(b) Foreign bodies in print, taken with the side camera

Figure 7.5: Foreign bodies in print

7.2.3. Pillowing

Pillowing occurs when the top layer is inadequately cooled, resulting in a surface characterised by uneven bumps or depressions, shown in Figure 7.6. Contributing factors include insufficient cooling of the top layer, reduced top shell thickness and an excessively high print speed for the top solid infill. Pillowing can be effectively identified using the nozzle camera, as this issue affects the top layer of a print. Parameters resulting into pillowing are provided in Table 7.3.

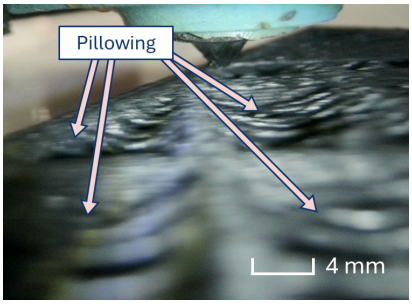


Figure 7.6: Example of pillowing, recognised by the depressions in the top printing layer

Test	Parameter	Value	Nominal
6	Top shell thickness	0.2 [mm]	0.7 [mm]
6	Top solid infill speed	50 [mm/s]	40 [mm/s]
6	Cooling top layer	max. 10%	max. 40%

Table 7.3: Printing parameters to induce pillowing

7.3. Support and Structural Issues

Support and structural issues are common challenges in FDM printing, impacting print quality and mechanical strength. These issues include failing supports, poor surface quality at support interfaces, weak infill, visible infill on the exterior and poor bridging.

7.3.1. Failing Support

Failing supports may arise when supports are poorly designed: e.g., too thin for the printer's resolution or too sparse in density or when isolated high towers are used as supports. Additional causes include excessive vibrations, high print speeds and insufficient bed adhesion, which can result from improperly prepared surfaces, such as missing polypropylene (PP) tape. PP GF30 requires PP tape for proper bed adhesion as the material only sticks well to itself. Side cameras can be used to detect these failing supports, as depicted in Figure 7.7. The print parameters utilised to achieve failing supports are provided in Table 7.4. However, upon further evaluation, it was found that failing supports bear a strong visual resemblance to spaghetti failure, covered in Section 7.5.4. The intertwined and scattered nature of dislodged supports closely mimics the characteristics of spaghetti failure, making it difficult to differentiate the two errors effectively. Due to this overlap, failing support was not considered a distinct error type and has instead been categorised alongside spaghetti failure.

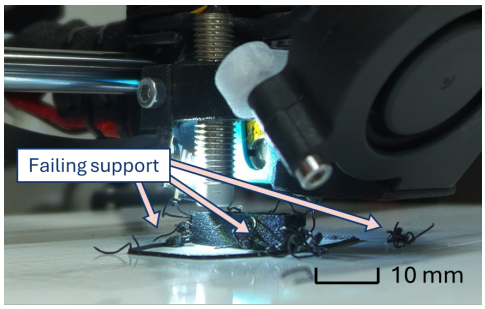


Figure 7.7: Example of failing support, characterised by its inability to sustain structures properly

Test	Parameter	Value	Nominal
7	Design of support	Isolated towers of 0.3 [mm]	-
7	Print speed support	50 [mm/s]	40 [mm/s]

Table 7.4: Printing parameters to induce failing support

7.3.2. Poor Surface Quality at Support

Poor surface quality at the interface between the print and its supports is a common issue. This type of defect occurs when the supports are poorly designed or misplaced, leading to improper contact with the printed part. Over-reliance on supports with a suboptimal pattern or excessive density can worsen the effect. If the nozzle temperature is too high, the printed part may adhere too strongly to the support, making post-processing challenging and damaging the surface finish of the part. Due to the complexity of visually identifying this defect before the supports are removed, this error was not covered in the fault detection system.

7.3.3. Weak Infill

Weak infill is another defect that can be identified using the nozzle camera. It occurs when the internal structure of the print lacks the necessary strength to provide adequate support for outer layers. Weak infill is typically the result of a nozzle temperature that is too low during infill printing, leading to poor layer bonding. Excessively high print speeds can also contribute to incomplete or inconsistent infill patterns, as can setting the infill overlap to zero. This introduces gaps between the infill and the outer wall. An example of weak infill is shown in Figure 7.8 on the next page, where the gaps and weak bonds are visible. Parameters for inducing weak infill are listed in Table 7.5 on the next page.

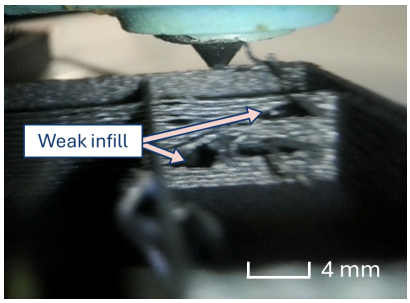


Figure 7.8: Example of weak infill, showing gaps in the infill structure

Test	Parameter	Value	Nominal
8	Nozzle temp	210 [°C]	235 [°C]
8	Print speed support	50 [mm/s]	40 [mm/s]
9	Overlap infill	0 %	10%

Table 7.5: Printing parameters to induce weak infill

7.3.4. Infill Showing on Exterior of Print

When infill structures become visible on the outer surface of a print, it can compromise the part's aesthetic quality and in some cases its functionality. This error often arises when the vertical shell thickness is insufficient to fully conceal the infill. Excessive infill overlap with the perimeters can push material outward, deforming the shell. Additionally, enabling the "infill before perimeters" setting can result in surface deformation. The reason is that the infill structure is printed before the outer shell, which can cause imperfections in the surface layer. The side cameras are suited for capturing this defect, as demonstrated in Figure 7.9. Testing parameters are outlined in Table 7.6. However, this error is often barely visible and can only be detected under specific lighting conditions, limiting the amount of usable data generated. This defect was not considered further for detailed analysis.

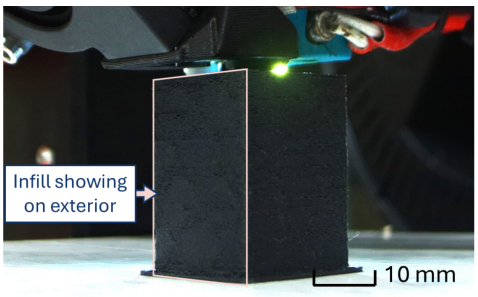


Figure 7.9: Example of infill on exterior, characterised by rough outer surfaces

Test	Parameter	Value	Nominal
10	Vertical shell perimeters	1	2
10	Overlap infill	100 %	40 [mm/s]
10	Infill before perimeters	True	False

Table 7.6: Printing parameters to induce infill on exterior

7.3.5. Poor Bridging

Bridging issues occur when long, unsupported horizontal sections of a print sag or fail. This defect is caused by inadequate cooling during bridge formation, excessive extrusion temperature or high print speeds, which do not allow the material to solidify. The side cameras are well-suited for observing poor bridging, as they can capture the deformation of unsupported spans, as shown in Figure 7.10. The specific conditions used to acquire data for this error are provided in Table 7.7.

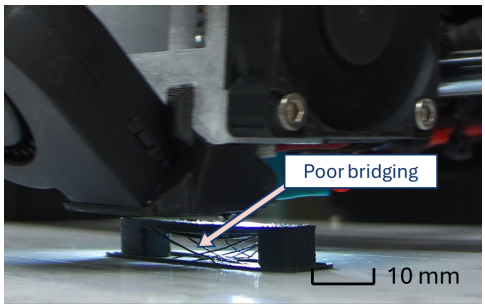


Figure 7.10: Example of poor bridging, where unsupported spans sag or fail

Test	Parameter	Value	Nominal
11	Design of part	Bridge of 20-50 [mm]	Supports
11	Max. fan speed	10%	40%
11	Bridge fan speed	10%	70%

Table 7.7: Printing parameters to induce poor bridging

7.4. Temperature and Cooling Issues

Temperature and cooling are critical factors in 3D printing. These can significantly influence the print quality. Incorrect temperatures or improper cooling settings can lead to a variety of errors such as stringing, overheating and overhang sagging. These arise due to imbalances in thermal management, either from excessive heat retention or insufficient cooling during the printing process.

7.4.1. Stringing

Stringing is a defect that appears as thin, unintended filament strands between separate parts of a print. This occurs when the retraction distance is insufficient during non-printing movements, as shown in Figure 7.11. Additionally, low travel speeds can increase stringing by giving molten filament more time to extrude during movement. Moisture within the filament worsens this problem, as the absorbed water turns to steam inside the nozzle, forcing material out. Both the nozzle camera and side cameras are effective for detecting stringing due to the visibility of fine strands across the print area. The parameters used to induce stringing are provided in Table 7.8.

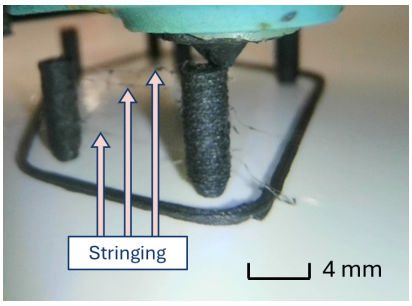


Figure 7.11: Example of stringing, recognised by a thin filament strand

Test	Parameter	Value	Nominal
12	Design of part	part requiring travelling	-
12	Retraction length	0.05 [mm]	0.8 [mm]
12	Travel speed	35 [mm/s]	180 [mm/s]
12	Environment	add moisture (water spray)	-

Table 7.8: Printing parameters to induce stringing

7.4.2. Overheating

Overheating refers to the excessive heat in printed layers, which causes the material to remain soft and to deform. Overheating typically arises from excessively high nozzle temperatures, inadequate fan cooling or excessively slow print speeds, preventing the printed layers from solidifying properly. During testing, it was observed that overheating often acts as a contributing factor to other errors, such as stringing, sagging or surface deformations, rather than being a standalone issue. Due to this interconnected nature, overheating was not explicitly categorised as an independent error in this system but was instead addressed indirectly through the detection of related defects. During the overheating tests, nozzle temperatures were set between 270–290°C, maximum fan settings was set to 5% and the print speed was reduced with 50% to replicate overheating effects and resulted in poor bridging, overhang sagging and pillowing.

7.4.3. Overhang Sagging

Overhang sagging occurs when layers extending beyond the previous layer's footprint begin to droop or deform, as visualised in Figure 7.12. This issue is particularly prevalent when overhang angles are too steep and when insufficient support structures are provided. It can be worsened by excessive nozzle temperatures, large layer heights or high print speeds, which reduce the time available for solidification. This defect is primarily captured using the side cameras, which provide a clear view of the sagging regions and the severity of the deformation. Testing for overhang sagging involves designing prints with challenging overhang angles and adjusting print parameters, as listed in Table 7.9.

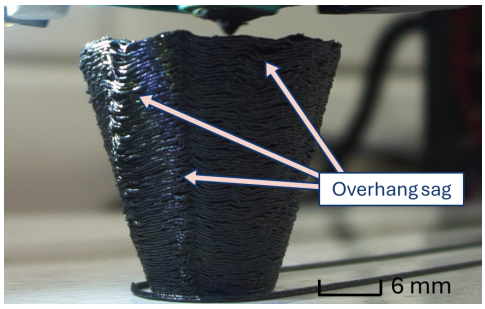


Figure 7.12: Example of overhang sagging, characterised by drooping material

Test	Parameter	Value	Nominal
14	Design of part	Create overhang	Supports
14	Nozzle temp.	255 [°C]	235 [°C]
14	Layer height	0.38 [mm]	0.2 [mm]
14	Print speed (all)	125%	100%

Table 7.9: Printing parameters to induce overhang sagging

7.5. Layer Alignment and Motion Issues

Layer alignment and motion issues refer to defects that arise from mechanical problems or misconfigurations in the printer. These errors occur due to improper maintenance, hardware malfunctions or environmental factors like vibrations. Common issues in this category include missing layers, shifted layers, ripples and echoes and spaghetti failure. These problems not only impact the structural integrity of the print but also compromise its aesthetic quality or even lead to complete failed prints.

7.5.1. Missing Layer

A missing layer occurs when a printer fails to deposit material for one or more layers during the printing process, resulting in gaps or weakened sections in the print. While missing layers may seem distinct, their characteristics are often similar to delamination or under extrusion. The lack of unique visual traits for missing layers complicates their identification. Missing layers are not explicitly addressed in the current system.

7.5.2. Shifted Layer

Shifted layers occur when the alignment of one or more layers deviates from the intended position. This leads to a visibly distorted print, as shown in Figure 7.13. This defect results from mechanical problems, such as loose belts, faulty stepper motors or physical obstructions during the print. Extreme warping can also induce this effect by exerting lateral forces on the printed part. Detection of shifted layers is achieved using side cameras, which provide a clear perspective of misaligned sections. To simulate this defect during testing, intentional interventions were performed, such as obstructing the head block and designing a misaligned part, summarised in Table 7.10.

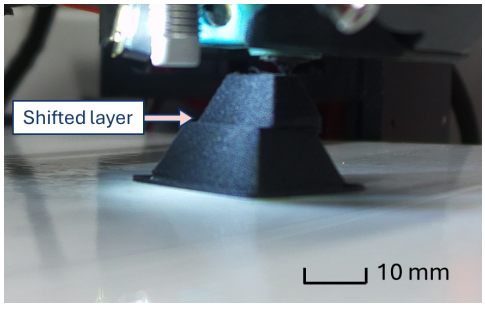


Figure 7.13: Example of shifted layer, showing a misalignment of layers

Test	Parameter	Value	Nominal
15	Design of part	Make a shifted layer in the design	-
16	Printer	Obstruct head block	-

Table 7.10: Printing parameters to induce shifted layer

7.5.3. Ripples and Echoes

Ripples and echoes refer to wave-like patterns on the surface of a print, typically caused by mechanical vibrations or backlash in the printer's motion. These defects occur when the printer's movement is not smooth, resulting in repeated distortions that mimic the shape of mechanical oscillations. This error is challenging to induce in well-maintained printers operating in stable environments. Moreover, modern printers with rigid frames and precise motion systems are less prone to such errors. Due to the rarity in well-maintained setups, ripples and echoes were excluded from the scope of this detection system.

7.5.4. Spaghetti Failure

Spaghetti failure is a severe defect characterised by chaotic filament strands, as depicted in Figure 7.14. This failure typically occurs when the print becomes detached from the build plate, causing the printer to extrude filament into the air. As a result, the filament accumulates in a random, unstructured manner. Spaghetti failure is often a consequence of other errors, such as poor bed adhesion, warping or improper Z-offset calibration. Once this failure begins, recovery is generally impossible and the print must be restarted. Both the nozzle camera and side cameras are capable of detecting this defect due to the distinct visual clutter it creates. The parameters used to induce spaghetti failure during testing are outlined in Table 7.11.

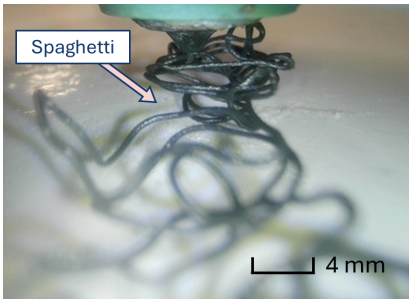


Figure 7.14: Example of spaghetti failure, characterised by chaotic strands

Test	Parameter	Value	Nominal
17	Nozzle temp.	260 [°C]	235 [°C]
17	Z-offset	0 [mm]	- 0.34 [mm]

Table 7.11: Printing parameters to induce spaghetti failure

7.6. Layer Adhesion Issues

Layer adhesion issues arise when layers of material fail to bond correctly, either to each other (inter-layer adhesion) or to the printer bed. These issues manifest in a variety of forms, such as first layer problems, warping, bed separation, elephant foot and delamination. Poor layer adhesion can result from improper print settings, environmental factors or material inconsistencies. They play a critical role in determining the structural integrity and quality of the print.

7.6.1. First Layer Problems

First layer issues include dimensional inaccuracies, poor bed leveling and irregular shapes such as circles that are not perfectly round. These issues are challenging to detect using vision-based methods alone because the system cannot distinguish between intentional design choices (e.g. ovals) and unintended dimensional deviations. Therefore, first layer problems were excluded as a stand-alone error in this system. However, these issues could potentially be addressed by simultaneously analysing the G-code during the print process, enabling comparisons between the intended and actual printed part. Integrating G-code analysis would allow for real-time evaluation of dimensional accuracy and geometric conformity, although this remains outside the current scope of the current research.

7.6.2. Warping

Warping occurs when the corners or edges of a printed object lift or curl away from the build plate over time, primarily due to uneven cooling and thermal stresses. This deformation is observed in prints with sharp corners, as these areas are prone to stress accumulation during cooling. Poor bed levelling, a bed temperature that is too low and a large Z-offset are key contributors to warping. The error is detectable using side cameras, as shown in Figure 7.15. Table 7.12 provides the print parameters used to simulate this issue.

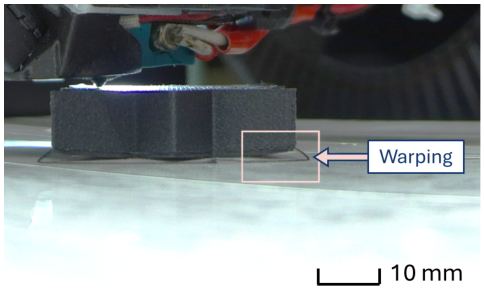


Figure 7.15: Example of warping, where sharp corners curl away from the bed

Test	Parameter	Value	Nominal
18	Nozzle temp.	260 [°C]	235 [°C]
18	Z-offset	0	0.34 (calibrated)
18	Bed temperature	0 [°C]	40 [°C]

Table 7.12: Printing parameters to induce warping

7.6.3. Bed Separation

Bed separation occurs when the print fails to adhere to the build plate, leading to partial or complete detachment. This error can be identified using side cameras and the nozzle camera, as shown in Figure 7.16. Common causes include poor bed levelling, excessive print speed or an incorrect nozzle height. In this study, the Z-offset was adjusted to induce bed separation intentionally. When these settings did not cause full detachment, the part was manually disconnected from the bed during printing. Bed separation is considered an extreme form of warping, as the entire print may eventually dislodge from the build plate. Subsequent analysis revealed that bed separation leads to spaghetti failure. Thus, this error is no longer considered a separate category. Print parameters to induce bed separation are summarised in Table 7.13.

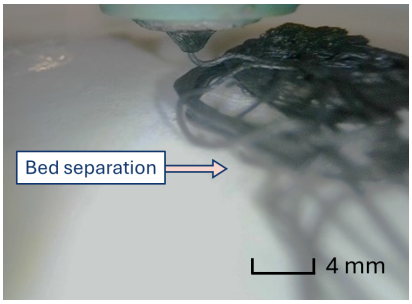


Figure 7.16: Example of bed separation, showing the part that is detached from the bed

Test	Parameter	Value	Nominal
19	Z-offset	0	0.34 (calibrated)
19	Print speed (all)	150 %	100 %

Table 7.13: Printing parameters to induce bed separation

7.6.4. Elephant Foot

Elephant foot refers to the slight bulging of the bottom layers of a print, causing the base to be wider than the rest of the object. This error occurs when the nozzle is too close to the bed (a Z-offset that is too small) or when the bed temperature is excessively high. The effect is subtle but can compromise dimensional accuracy and aesthetic quality, as shown in Figure 7.17. To replicate this error, the setting designed to prevent elephant foot was disabled. The associated print parameters are provided in Table

7.14. Despite purposeful induction, the error was found to be too minor for reliable detection by the neural network system. As such, elephant foot was excluded as error type.

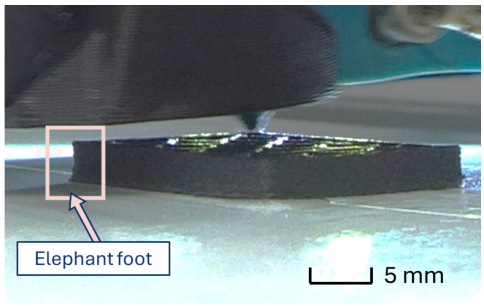


Figure 7.17: Example of elephant foot, characterised by the first layer bulging out

Test	Parameter	Value	Nominal
20	Nozzle temp.	260 [°C]	235 [°C]
20	Z-offset	-0.50 [mm]	-0.34 [mm]
20	Bed temperature	0 [°C]	40 [°C]

Table 7.14: Printing parameters to induce elephant foot

7.6.5. Delamination

Delamination is a form of inter-layer adhesion failure in which cracks or splits appear between layers of the print. Unlike warping, which affects the base of the print, delamination occurs within the object itself and is typically observed in tall prints or objects with sharp corners, as shown in Figure 7.18. This issue arises due to factors such as excessive cooling, large layer heights, external airflow, high print speeds and insufficient extrusion in specific areas. To simulate delamination, the flow rate was reduced for some corners of the part. Other parameters were adjusted as summarised in Table 7.15. Side cameras were able to capture this error accurately, highlighting its potential for detection within the proposed system.

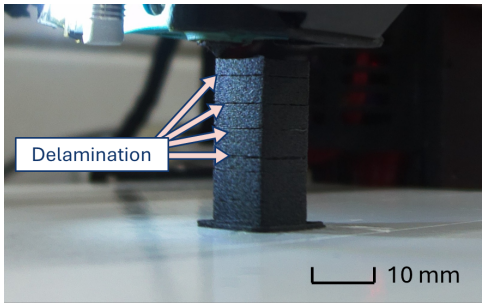


Figure 7.18: Example of delamination, showing inter-layer separation

Test	Parameter	Value	Nominal
21	Design of part	Tall object with sharp corners	235 [°C]
21	Max. fan speed	100%	40%
21	Print speed (all)	150%	100%
21	Flow rate few corners	10%	100%

Table 7.15: Printing parameters to induce delamination

8| Dataset

The design of the dataset is a critical step in developing an effective fault detection system for 3D printing. This chapter outlines the considerations taken in selecting shapes and infills, structuring the dataset for training and validation and defining the external validation set. These steps ensure that the dataset is diverse, balanced and resulting in a model that can generalise well.

8.1. Shapes and Infills

A diversity of shapes and infills is crucial to create a robust dataset capable of generalising across various 3D printing scenarios. This ensures that the model learns to identify faults based on the errors themselves, rather than associating specific shapes or infills with certain errors. The dataset consists of 16 base shapes, depicted in Figure 8.1. For some shapes, variations were introduced, such as different scaling factors, to accommodate a specific error. These variations are not shown individually in the figure, as they primarily serve to test edge cases or refine model performance. Specific features were required for certain errors, limiting randomisation of shapes. For instance: poor bridging requires geometries containing bridges, which are only achievable with shapes 5, 6, 7, 8 and 14.

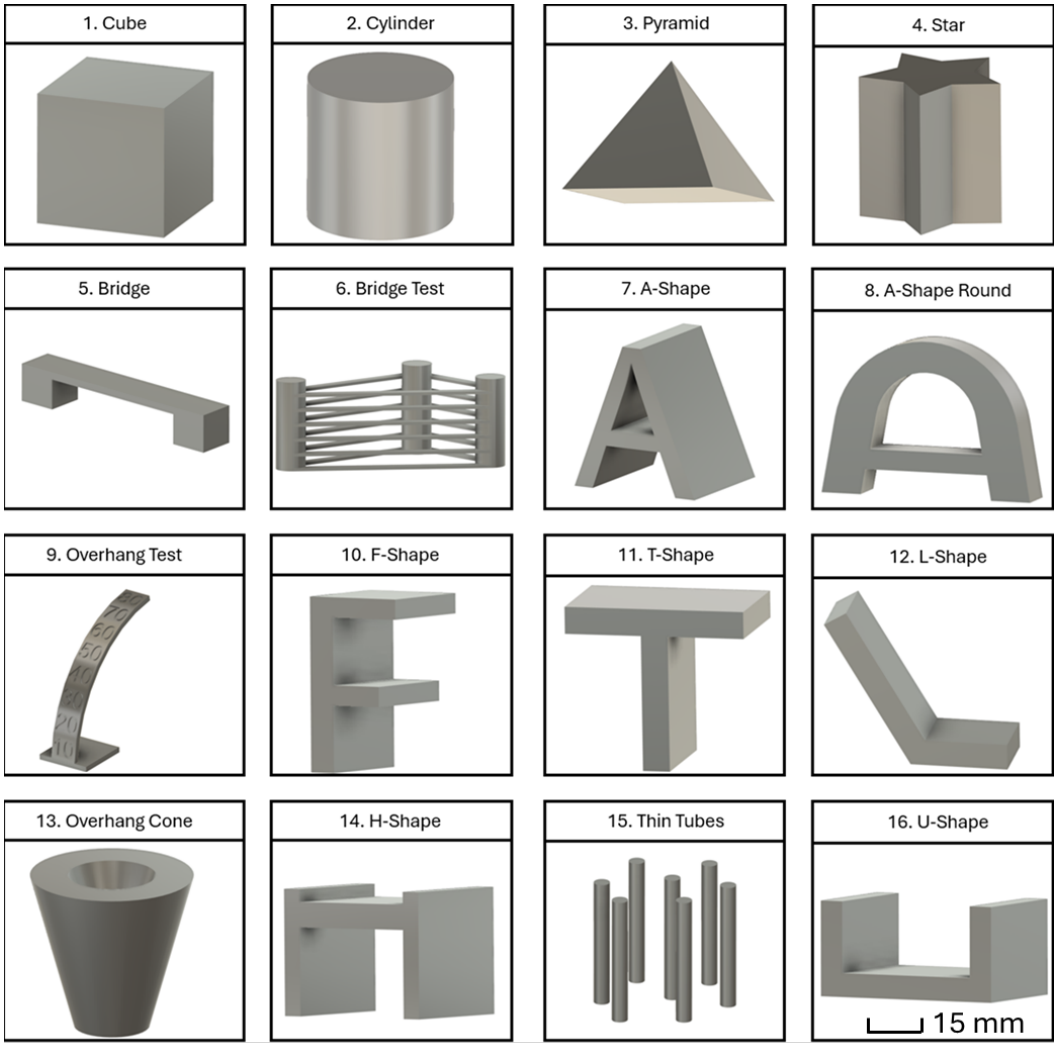


Figure 8.1: Base shapes used for training, validation and test set

Infills, the internal structures of 3D prints, were varied to increase the diversity of the dataset. These infills, selected as percentages of density, play a critical role in 3D printing by influencing strength, material usage and print time. For this study, infill percentages were chosen within a range of 2% to 50%, using integer values only. The boxplot shown in Figure 8.2 illustrates the distribution of these infill percentages, with a median of 15%. Notably, 15% represents the default setting in PrusaSlicer, the slicing software used in this work. The plot also shows two tests with infill percentages of 40% and 50%, demonstrating the inclusion of high infill values. This distribution was intentionally designed to reflect real-world printing scenarios and to capture a wide range of potential variations.

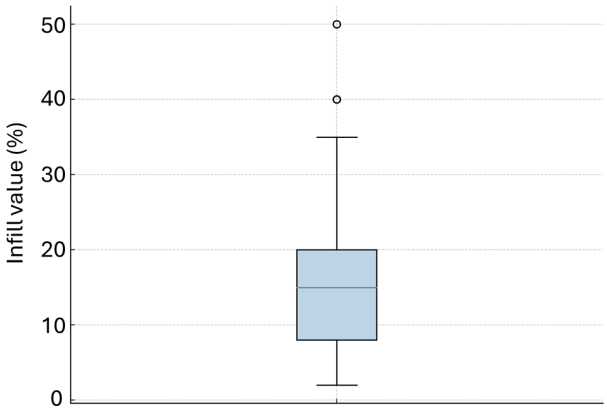


Figure 8.2: Boxplot distribution of infills (%), including infills between 2-50%, with a median of 15%

A detailed overview of shapes and infills, categorised by error type, is provided in Table 8.1, ensuring transparency and reproducibility. The table also identifies any shape-specific variations.

Error Type	Test	Shape	Infill (%)
Good	0	Cube	5
	0	Cylinder	15
	0	6 Low cubes	10
	0	F-shape	30
	0	Pyramid	10
	0	Bridge	10
	0	Bridge test	20
	0	Overhang cone	25
	0	U-shape	10
	0	Overhang test	20
Over Extrusion	1	Cylinder	15
	1	Cube	25
	2	Pyramid	40
Under Extrusion	3	Pyramid	5
	3	6 Low cubes	50
	3	Cube	30
	3	Low star	5 / 10
	3	Low oval	20
	4	Low cylinder	25
Foreign Body	5	Star	25
	5	A-shape	5
	5	Cylinder	40
	5	Bridge	5
	5	Pyramid	15
	5	T-shape (up)	8
	5	Cube	5
Pillowing	6	Low cube	4
	6	Cylinder	5
	6	Star	4
Weak Infill (%)	8	Low star	8
	8	Triangle pyramid	10
	9	Tall cube	15
Poor Bridging	11	Bridge	40
	11	Bridge test	10
	11	Bridge oriented	20
	11	A-shape	10
	11	A-shape block	5
	11	H-shape	30

Error Type	Test	Shape	Infill
Stringing	12	A-shape block	15
	12	H-shape	20
	12	Thin tubes	15 / 20
	12	U-shape	10
Overhang Sag	14	Overhang cone	5
	14	2 Overhang cones	3
	14	L-shape	10
	14	Overhang test	15
	14	Overhang cone (sq)	10
	14	Overhang cylinder	8
	14	4 Overhang tests	18
Shifted Layer	15	Cube	20
	15	Cylinder	15
	15	Pyramid	35
	16	A-shape	10
	16	F-shape	20
Spaghetti	7	T-shape	10
	7	H-shape	35
	7	F-shape	25
	17	Cube	20
	17	Cylinder	5
	19	L-shape	20
	19	Overhang cone	30
Warping	18	Low cube	10
	18	Pyramid	12
	18	Low star	8
	18	Triangle pyramid	15
	18	A-shape block	30
Delamination	21	Tall cube	5
	21	Tall star	2
	21	Tall triangle pyramid	12
	21	A-shape block	8
	21	A-shape round	3
	21	T-shape up	15
	21	Tall H-shape	3

Table 8.1: Shapes and infills per error type

8.2. Data Split

A proper division of the dataset is important to develop and evaluate the quality of the model. Splitting the data into training, validation and testing subsets ensures that the model learns from one portion of the data, tunes its parameters using another and is tested on unseen examples. Figure 8.3 explains visually how splitting of data took place, from labelling and balancing to the final division into subsets.

Initially, all images from both the nozzle and side cameras were labelled according to their respective error classes. To achieve a balanced dataset, the number of images per class was equalised. For the nozzle camera, each error class was limited to a maximum of 2,500 images, while for the side camera, this number was capped at 2,000. When an error class exceeded these limits, random images were removed to reduce the dataset to the target size. These dataset sizes were determined through a trial-and-error process, evaluating the model's performance at increasing dataset sizes until sufficient accuracy was obtained [63, 64].

Once the dataset was balanced, it was randomly split into three subsets: 70% training, 20% validation and 10% testing data. This split was chosen as it is a widely accepted standard, ensuring that sufficient data is allocated for training while retaining enough samples for validation and testing [65]. Importantly, the split was performed independently for each class to maintain balance across the subsets. This random splitting ensures that every class is proportionally represented in the training, validation and testing datasets. The randomisation process in Python is used to split the images into three sets.

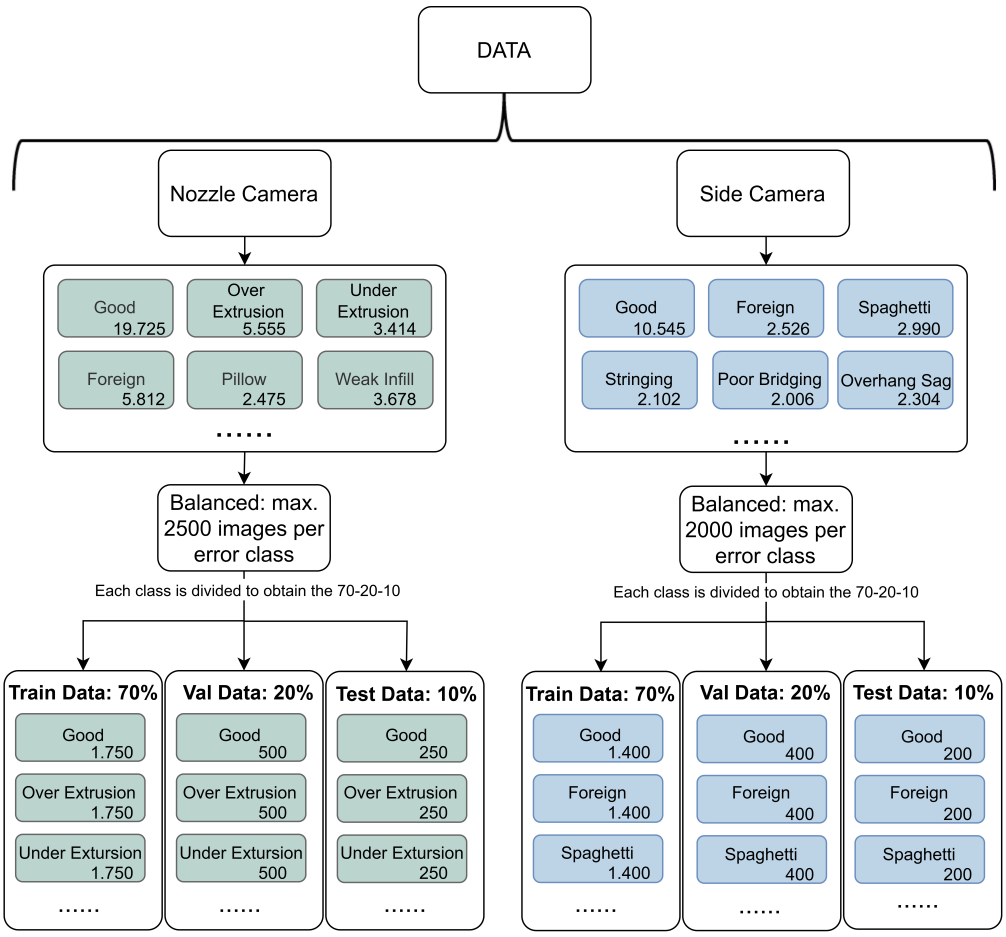
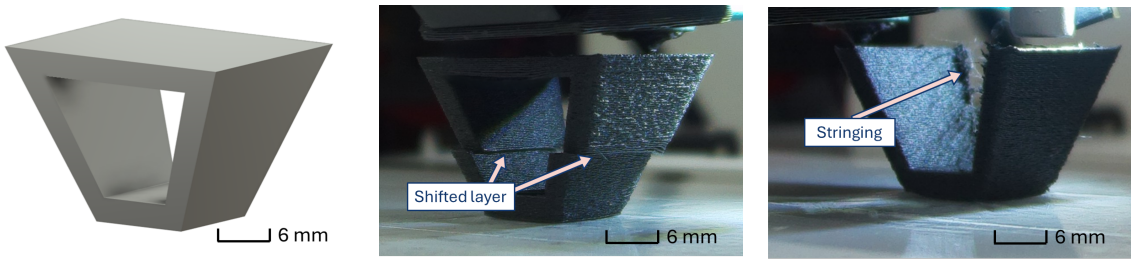


Figure 8.3: Process of data splitting: balancing, and splitting in training, validation and test data

8.3. External Validation Set

Since the training, validation and test sets were randomised from the same dataset, this approach alone does not fully assess the model's ability to generalise to unseen data, containing other patterns. To address this limitation, an external validation dataset was introduced as an independent mechanism for assessing the model's classification capabilities. This external set tests the model's ability to identify errors in entirely new shapes, validating it performs beyond the data it was trained on.

To create this dataset, a custom 3D shape was designed to incorporate all possible error types, ranging from overhang sagging to poor bridging, stringing, pillowing and more. This shape was printed with all errors, enabling a systematic evaluation. Figure 8.4 illustrates this process, showing the CAD model of the external validation shape alongside images captured by the side camera that demonstrate shifted layers and stringing errors.



(a) CAD model of shape **(b)** Shifted layer of the external validation **(c)** Stringing of the external validation

Figure 8.4: External validation set shape and two examples of errors

The external validation dataset is inherently unbalanced, with varying numbers of images per class. This imbalance arises because each test was conducted only once and certain error types are visible only under specific conditions. For instance, pillowing is exclusively observed on the top layer, limiting the number of corresponding images. The distribution of images across classes is listed in Table 8.2.

Error Type	# Nozzle Images	#Side Images
Good	2742	397
Over Extrusion	475	-
Under Extrusion	139	-
Foreign Body	364	548
Pillowing	173	-
Stringing	377	62
Spaghetti	544	387
Weak Infill	19	-
Poor Bridging	-	105
Overhang Sag	-	224
Shifted Layer	-	148
Warping	-	251
Delamination	-	292
Unclear	465	291
Not Printing	433	102
Total	5731	2807

Table 8.2: External dataset size per class

9| Test Plan

This chapter describes the step-by-step approach to conduct tests for data collection. The outlined process ensures that the testing is carried out systematically, producing reliable and high-quality data for model training and evaluation. For each test there are four phases to be executed to gather data: preparation, setup, execution and post-processing. This is graphically presented in Figure 9.1.

The first step involved the preparation. CAD models were designed to create the necessary shapes for testing. Using PrusaSlicer, these models were converted into printer-readable files or G-codes. These G-codes were designed to introduce specific print errors, as detailed in Chapter 7. After generating the G-code files, they were uploaded to OctoPrint to facilitate remote control and monitoring of the 3D printer. Concurrently, the cameras were inspected to ensure they were operational and producing clear, focused images.

The second step involved the testing setup. The Python scripts responsible for capturing and saving images were updated to ensure data is stored in correct locations. For the side cameras, image data was directly saved onto a hard drive connected to the Raspberry Pi. In contrast, the nozzle camera stored its image data on an external drive via a laptop. This arrangement addresses power limitations, which are discussed in detail in Chapter 6.

Thirdly, the tests were conducted. Through OctoPrint, the printer was remotely started and the Python scripts for image capture were executed. Throughout the printing process, periodic monitoring is essential to ensure the print progresses as planned. Parameters may need adjustment if unexpected errors arise that are not aligned with the test objectives. Furthermore, the captured images were regularly reviewed to verify their quality and usability.

The final phase of the testing process involved post-test data management. Images captured by the nozzle camera were transferred to the same hard drive as the side camera data. The image quality was then reviewed to ensure all relevant details are accurately captured. Any unusable images were discarded. Once the dataset was organised, post-processing began, including labelling and sorting the images.

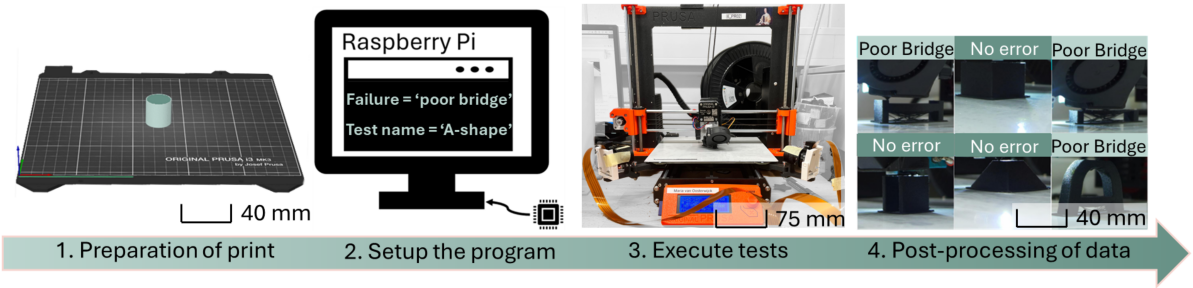


Figure 9.1: Graphical presentation of the test plan consisting of four steps: the preparation, the setup, the test execution and post-processing

10| Post-Processing and Data Augmentation

Post-processing and data augmentation are critical steps in preparing the dataset for neural network training. These steps ensure that the data is clean, organised and diverse. They enable the model to learn effectively while improving its generalisation capabilities. Post-processing encompasses tasks such as filtering, cropping and labelling the image data. Data augmentation involves applying transformations to the images. As a consequence, a more diversified dataset is generated.

10.1. Post-Processing

The initial step in post-processing involved manual filtering the collected images to eliminate corrupted, irrelevant or poorly captured data. This stage was essential to ensure the quality of the dataset, as unusable images can negatively impact the training process. Corrupted images, out-of-focus captures or those containing insufficient information were moved to a separate ”unclear” category for further inspection.

Once the images were filtered, they were cropped to optimise input size and reduce computational load during training. The cropping process was carefully designed to preserve all relevant information while minimising unnecessary details. Images from the nozzle camera were cropped from an initial resolution of 640x480 pixels to 440x320 pixels, while those from the side cameras were reduced from 4056x3040 pixels to 1600x1000 pixels. These dimensions ensured that the images retained essential visual information while significantly reducing their size, by 54.2% and 87.0% respectively for the nozzle and side images. Subsequently, during training using YOLOv8, the side camera images were rescaled to 512x512 pixels, facilitating faster computations.

Following the cropping process, a time-consuming manual sorting phase was undertaken to classify each image accurately. This task involved verifying that all images were correctly labelled according to their respective error types. If any were mislabelled, they were reassigned to the appropriate class. Additionally, two new categories were introduced: ”unclear” for images with indistinguishable details and ”not printing” for images captured during printer downtime. The final sorted dataset,detailing the number of useful images per error class and camera type before balancing, is summarised in Table 10.1.

Error Type	Nozzle Images
Good	19 725
Over extrusion	5 555
Under extrusion	3 413
Foreign body	5 811
Pillowing	2 474
Stringing	3 210
Spaghetti failure	5 545
Weak infill	3 677
Unclear	2 927
Not printing	4 492

Error Type	Side Images
Good	10 545
Foreign body	2 526
Stringing	2 102
Spaghetti	2 990
Poor Bridging	2 006
Overhang Sagging	2 304
Layer Shift	1 982
Warping	2 583
Delamination	2 572
Unclear	2 266
Not printing	1 349

Table 10.1: Number of images per class for nozzle and side cameras, before balancing

10.2. Data Augmentation

After sorting, the dataset was augmented to introduce variations and expand the amount of available training data. Data augmentation involves transforming the images while maintaining the original image labels. Horizontal flipping, rotation and colour adjustments, such as altering brightness, contrast and saturation, were applied. These transformations aimed to enhance the model's robustness by exposing it to diverse scenarios during training. Mosaic augmentation was utilised, where multiple training images were combined into a single training sample. This technique introduced additional variability and encouraged the model to recognise patterns in a more complex, dynamic environment.

Figure 10.1 illustrates examples of data augmentation. The original images are shown on the left, while their augmented counterparts appear on the right. These transformations improved the dataset's diversity, enabling the neural network to generalise more effectively.

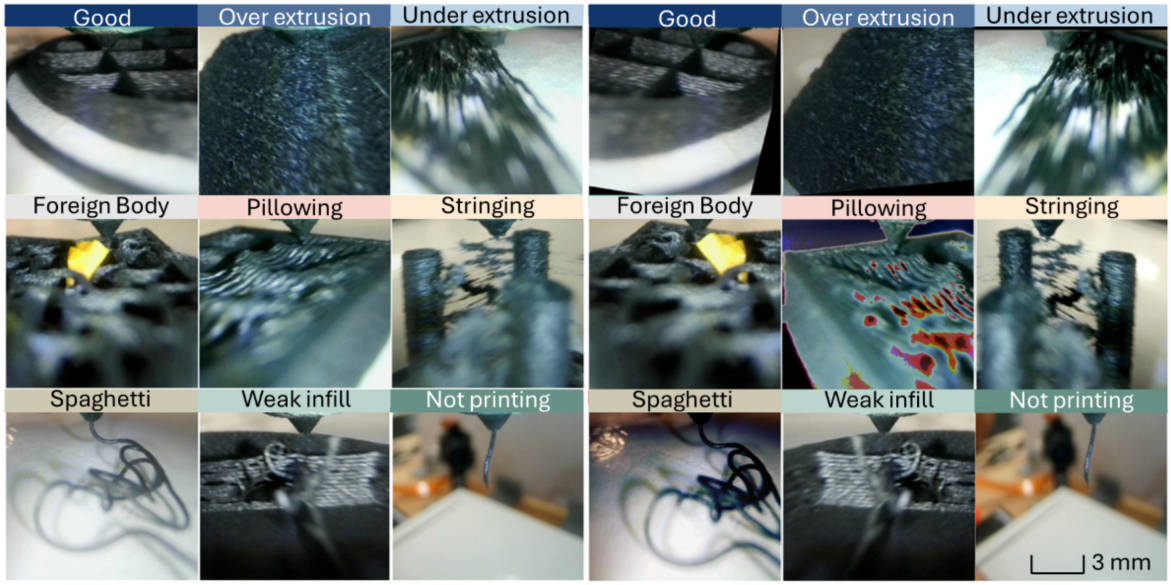


Figure 10.1: Data augmentation: left side shows the original images and the right side shows the augmented images

Upon the data acquisition process, encompassing the aforementioned steps, a diverse and high-quality dataset was created to represent a wide range of 3D printing faults. These efforts established a strong foundation for training, validating and testing the neural network models.

PART IV

Convolutional Neural Network Models

The development of a functional fault detection system for 3D printing relies on advanced machine learning techniques, particularly Convolutional Neural Networks (CNNs). CNNs have proven to be highly effective in image-based tasks due to their ability to automatically learn hierarchical features from raw pixel data. This Part introduces the fundamental concepts of CNNs, providing the necessary background to understand their application in fault detection for 3D printing.

Chapter 11 provides a general introduction to CNNs, covering the basics of neural networks, a more in-depth explanation of CNNs and an overview of the pre-trained YOLOv8 Classification model. Chapter 12 describes the training process for CNN models, discusses parameter tuning to optimise performance and explains the concept of transfer learning. Finally, Chapter 13 explores how to assess CNN models, detailing the calculation of accuracy, loss, precision, recall and F1 scores, as well as how to interpret these metrics.

11| Introduction to Convolutional Neural Networks

Machine learning is a subset of artificial intelligence that enables computers to learn and make predictions or decisions without being explicitly programmed [66]. At its core, machine learning uses algorithms that analyse and identify patterns in data, enabling models to improve their performance. Neural networks, a specialised subset of machine learning, are inspired by the structure and function of the human brain [67]. These networks 'learn' by adapting their internal parameters based on the data they process, making them powerful tools for solving complex problems. In Section 11.1, the working principle of NN are explained. Section 11.2 explains Convolutional Neural Networks (CNN). CNNs are specialised types of NN, capable of processing images.

11.1. Neural Networks

Neural networks are computational frameworks that mimic the way the human brain processes information [67]. They consist of three primary types of layers: the input layer, hidden layers and the output layer [68]. The input layer serves as the entry point for raw data, such as images or numerical values and passes this data to the subsequent layers. The hidden layers act as intermediaries, transforming the input data into abstract representations by applying a series of mathematical operations. These transformations allow the network to learn intricate patterns within the data. Finally, the output layer generates the model's predictions, whether they are classifications, regressions or any other form of output. Figure 11.1 illustrates the core components of a neural network [69]. It is a simplified representation of what are often complex and multi-layered architectures. In real-world applications, modern networks may contain hundreds or even thousands of layers.

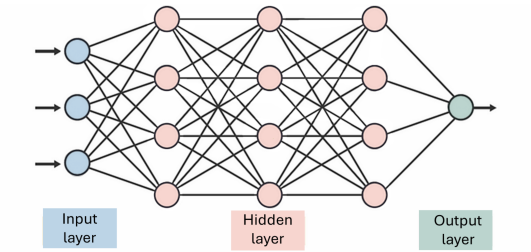


Figure 11.1: Neural network structure consisting of input, hidden and output layers (adapted from [69])

The individual neurons are the fundamental units within each layer. Each neuron operates as a function approximator, receiving inputs from the previous layer and combining them through weighted connections [70]. These weights, denoted by w_{ij}^{l} , determine the strength of the influence each input has on the neuron. The neuron also incorporates a bias term, b_j^l, which enables it to shift its activation independently of the input. This process is mathematically expressed in Equation 11.1, where z_j^l represents the weighted sum of inputs for neuron *j* in layer *l*, while $a_i^{(l-1)}$ denotes the activations from the previous layer and n represents the amount of neurons.

$$z_j^l = \sum_{i=1}^{n} \left(w_{ij}^l \cdot a_i^{(l-1)} + b_j^l \right) \tag{11.1}$$

Once the weighted sum is computed, it is passed through an activation function. This function introduces non-linearity into the network. This non-linearity allows the model to learn complex patterns [71]. Without these functions, the network would behave as a linear system, regardless of the depth of its layers. Each neuron in the network applies an activation function to the weighted sum of its inputs to determine its output. Common activation functions include: ReLu, sigmoid, tanh and SiLu activations. ReLU (Rectified Linear Unit) outputs the input if positive; otherwise, it returns zero. This is widely used due to its simplicity and effectiveness. The sigmoid activation outputs values between 0 and 1, often used for binary classification problems. Tanh works similarly to the sigmoid in providing non-linearity but offers a symmetric output range: tanh outputs values between -1 and 1, providing stronger gradients for inputs closer to zero. The SiLU activation function (Sigmoid-Weighted Linear Unit) combines sigmoid and ReLu, offering smooth and efficient gradient-based learning. These activation functions are plotted in Figure 11.2.

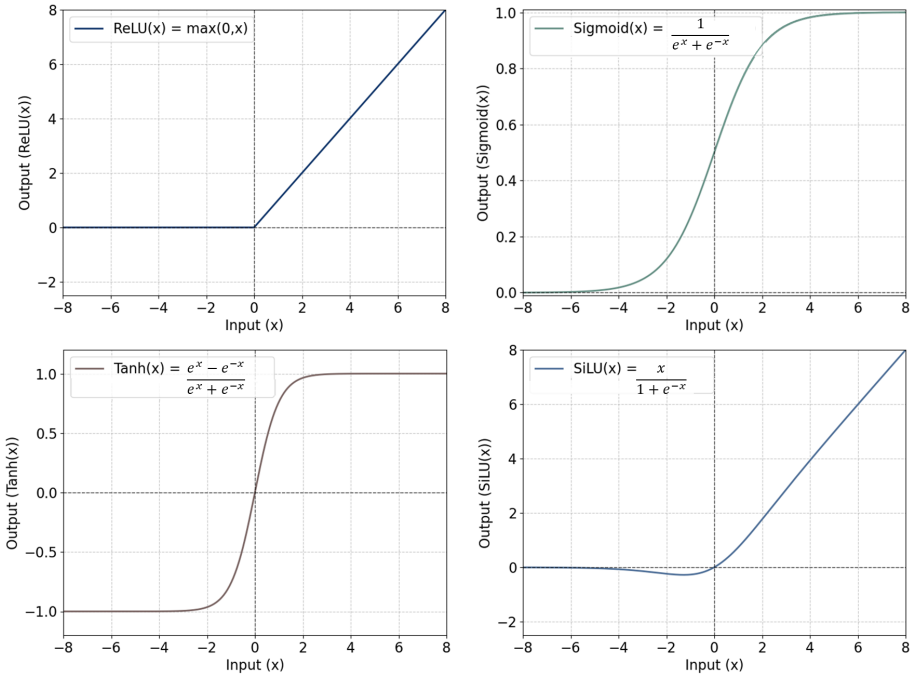


Figure 11.2: Plots of ReLu, Sigmoid, Tanh and SiLu Activation Functions, where x is the weighted sum input and y is the output after non-linearity is introduced

This process is analogous to how biological neurons function. In the human brain, neurons receive inputs through dendrites, process them in the cell body and send outputs via axons to other neurons. Similarly, in neural networks, neurons receive data, process it through mathematical operations and pass the result to the subsequent neurons. This resemblance is visualised in Figure 11.3 [72].

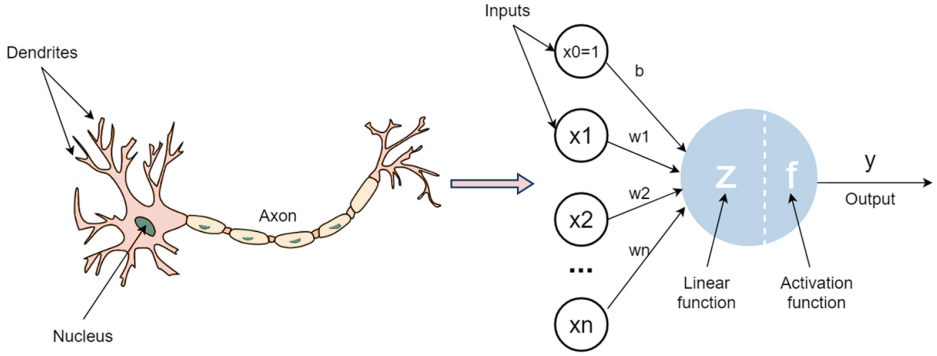


Figure 11.3: Neurons in humans (left) and within a neural network (right) (adapted from [72])

11.2. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a specialised type of neural network primarily designed for processing grid-structured data, such as images. In computer systems, images are represented as grids of numerical values (matrices). A single image is composed of three matrices, each representing the pixel intensities of the RGB colours (Red, Green or Blue). CNNs use the spatial relationships within this grid-structured data. That is why they are highly effective for tasks such as image classification, object detection and segmentation. Unlike traditional neural networks, which treat input data as a flat vector, CNNs maintain the spatial grid structure of the input. This approach allows the network to detect patterns like edges, shapes and textures. For the 3D printing fault detection system, a CNN is employed to classify images into either a ”good” category or a certain ”error” class.

The fundamental building block of a CNN is the convolutional layer. This layer applies a set of learnable filters (or kernels) to the input image through a mathematical operation called convolution. Each filter is a small matrix (e.g., 3x3 or 5x5) that slides over the input image, computing the dot product between the filter and the corresponding region of the image. The result is a feature map, which highlights specific patterns, such as edges or textures, detected by the filter. [73]

Figure 11.4 shows the convolutional operation happening to an image. The kernel of size 2x2 slides over the image, with step size or stride 1 and performs the dot product of the respective matrices. The results are then stored in a feature map, as shown on the right side of the image. One important thing to note is that by performing this operation, the image gets smaller: the input image is 8x8 pixels, while the feature map produced is reduced to 7x7 pixels. When taking a larger stride, the image is reduced even further. Zero padding is often applied to prevent this reduction in dimensions. Zero Padding is a technique used to maintain the spatial dimensions of the input by adding extra rows and columns filled with zeros around the edges of the original image. This ensures that the output feature map retains the same size as the input. Alternatively, same padding can be applied. Then, rows and columns are added with the same number as outer edges of the image.

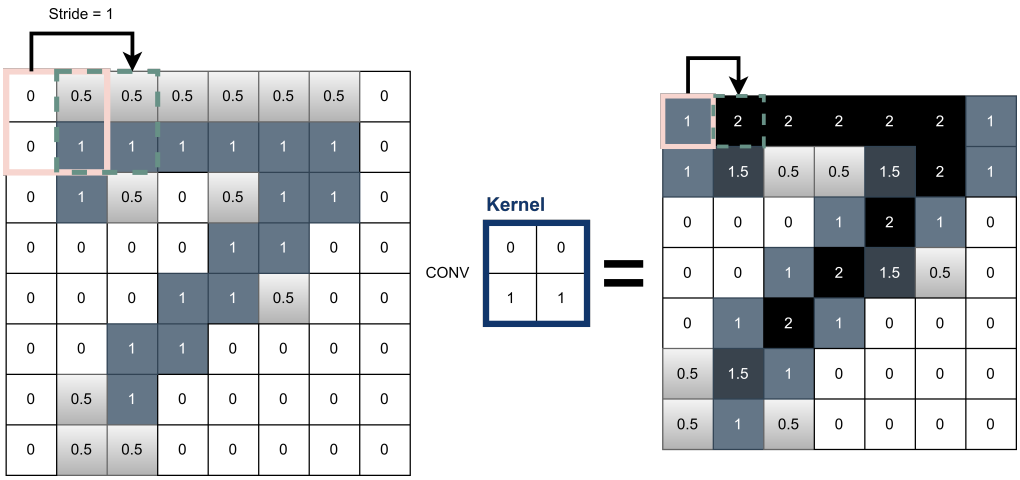


Figure 11.4: Example of a convolution layer, consisting of 1 kernel (2x2)

Pooling layers are another key component of CNNs, designed to reduce the spatial dimensions of feature maps, which helps decrease the computational complexity and prevents overfitting. The two most common pooling operations are: max pooling and average pooling. In max pooling, the maximum value from a specific region in the feature map is selected, preserving the most significant feature in that region. In average pooling the average of all values within a region is computed, providing a smoother representation. Figure 11.5 shows an example of max pooling and average pooling. [73]

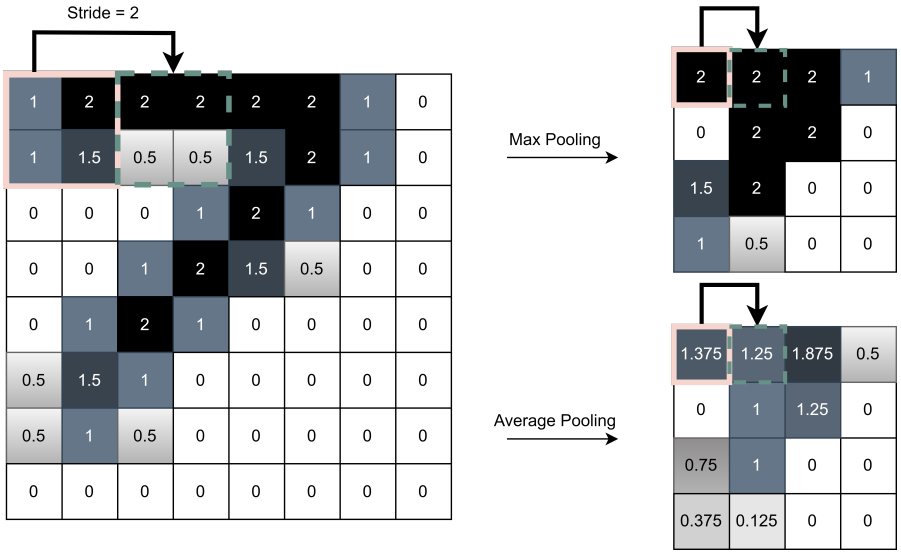


Figure 11.5: Example of max pooling (top) and average pooling (bottom)

After multiple convolutional and pooling layers, the feature maps are typically transformed into a one-dimensional vector using a flatten layer. This step is necessary because the final stage of a CNN, which involves classification or regression, requires the input to be in vector format. Flattening essentially converts the spatial structure of the data into a format suitable for fully-connected layers.

The fully-connected (dense) layers are typically placed at the end of the network. These layers take the flattened vector as input and use it to make predictions. Each neuron in a fully-connected layer is connected to every neuron in the previous layer, enabling the network to learn global patterns that influence the final classification or regression task. Figure 11.6, depicts an example of the key components in CNN: two convolutional layers, two pooling layers, a flattening layer and fully connected layers [74].

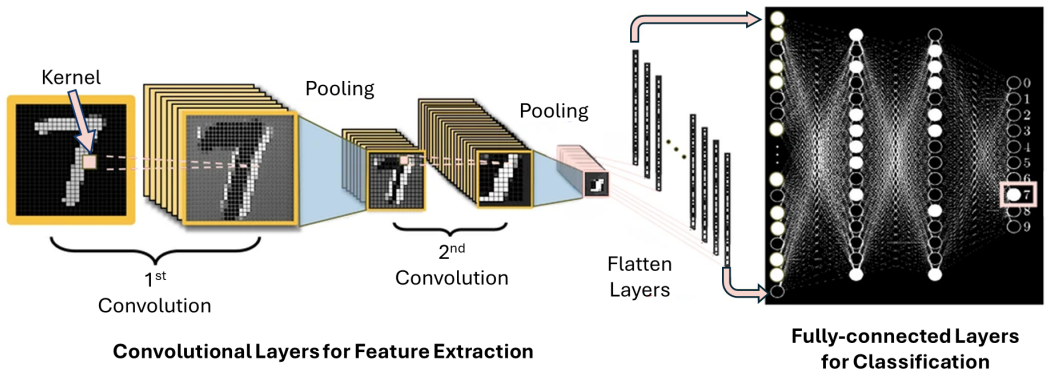


Figure 11.6: Example of CNN showcasing convolutional, pooling and fully connected layers (adapted from [74])

11.3. YOLOv8 Classification Model

The fault detection system developed for this thesis relies on the pre-trained YOLOv8 classification model as the foundational framework for image classification tasks. YOLOv8 is a cutting-edge pre-trained model known for its efficiency and high accuracy, particularly in object detection tasks. For classification and object detection purposes, YOLOv8 significantly outperforms other pre-trained models such as ResNet50 and VGG16 [75–77]. Object detection goes beyond classification as it aims to locate specific features within an image. Although object detection is beyond the scope of this thesis, it remains a promising direction for future research. This is because object detection could enable fault correction by identifying the precise locations of faults. For the current work, YOLOv8's classification functionality has been utilised to reliably distinguish between a good state and different error types in 3D printing. The effective classification and the potential for object detection established YOLOv8 as a good choice for the real-time detection system during 3D printing.

The architecture of the YOLOv8 classification model is composed of distinct building blocks, each designed to fulfil a specific role in the process of feature extraction and classification. These blocks work in harmony to achieve a balance between computational efficiency and accuracy. The three main blocks include: the convolution block, the C2f block and the classify block. A simplified representation of the YOLOv8 classification model is shown in Figure 11.7 on the next page.

The Convolution Block is the fundamental unit of the YOLOv8 architecture. It consists of three parts: a 2D convolutional layer, batch normalisation and SiLU activation. The 2D convolutional layer is responsible for extracting spatial features from the input images. This allows the model to capture patterns such as edges, textures and shapes. Batch normalisation ensures stable and efficient training by normalising the intermediate outputs so that all outputs are within the range [0, 1]. This helps stabilising the learning process and accelerates convergence. The SiLU activation introduces non-linearity, while maintaining smooth gradients for improved optimisation.

The C2f Block or "cross-stage partial connections with bottleneck block" is designed to improve feature extraction efficiency. Unlike a standard Convolution Block, the C2f Block uses shortcut connections to reuse features from earlier layers. It consists of two bottleneck blocks that reduce the dimensionality of the feature maps before expanding them back to their original size. This process helps the network to focus on key features while minimising computational complexity. This makes the model both efficient and lightweight.

The Classify Block is the final stage of the YOLOv8 classification model, specifically tailored for generating the classification output. It begins with a Convolution Block that further processes the extracted features. A max pooling layer is then applied to downsample the feature maps. Dropout is introduced to mitigate overfitting by randomly deactivating a fraction of neurons during training. Finally, the fully connected layer integrates the processed features into a compact vector and produces the classification result, assigning the input image to a specific output class.

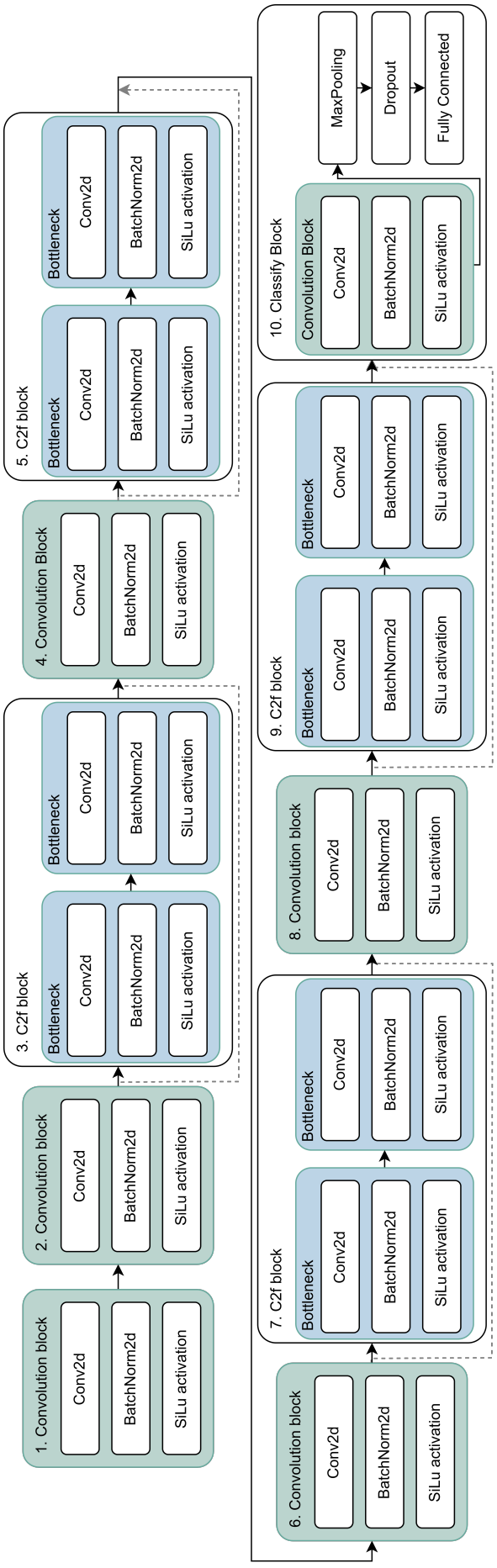


Figure 11.7: Simplified architecture of the YOLOv8 classification model, comprising convolution blocks, C2f blocks and a classify block

12| Model Training, Parameter Tuning and Transfer Learning

This chapter describes the training process of a CNN model, focusing on the mechanics of parameter optimisation and validation. Techniques for improving the model's performance through parameter tuning and transfer learning are explored.

12.1. Training Process

The objective of training a neural network is to minimise the error by optimising its parameters, which include the weights and biases of the model. This error is measured using a loss function. This quantifies the difference between the predicted outputs and the actual target values. The optimisation process involves backpropagation, where the model computes the loss and propagates this error backward through its layers. These calculations begin at the output layer and propagate backward through the network.

The weights and biases are updated iteratively using gradient descent, an optimisation algorithm that adjusts the parameters to minimise the computed loss [78]. Specifically, the weights and biases are updated by subtracting the gradients of the loss function, scaled by the learning rate (α). This learning rate determines the step size of the updates and is described mathematically by Equations 12.1 and 12.2, where Δw_{ij}^{l} and Δb_{j}^{l} represent the updates to the weight and bias and $\frac{\partial \text{loss}}{\partial w_{ij}^{l}}$ and $\frac{\partial \text{loss}}{\partial b_{j}^{l}}$ denote the gradients of the loss function with respect to the weight and bias, respectively. Various optimisation techniques have been developed to enhance gradient descent by improving convergence and adapting learning rates dynamically. One commonly used optimisation technique is Adam optimiser. The Adam optimisation algorithm is employed to enhance the computational efficiency of the model [79].

$$\Delta w_{ij}^{l} = -\alpha \cdot \frac{\partial \text{loss}}{\partial w_{ij}^{l}} \tag{12.1}$$

$$\Delta b_{j}^{l} = -\alpha \cdot \frac{\partial \text{loss}}{\partial b_{j}^{l}} \tag{12.2}$$

Training typically occurs in batches rather than processing the entire dataset at once. Each batch, a subset of the training data, is passed through the model during a single forward and backward pass. This technique, called mini-batch gradient descent, enhances computational efficiency and enables faster convergence. For example, with a batch size of 16, 16 images are processed at once. The weights and biases are updated accordingly before proceeding to the next batch. This continues until all data in the training set is processed.

An epoch refers to one complete pass through the entire training dataset. As training progresses over multiple epochs, the model refines its ability to make accurate predictions. However, it is crucial to monitor the model's performance to avoid overfitting. Overfitting refers to the model becoming too specialised in the training data and failing to generalise to unseen data. This overfitting is graphically presented in Figure 12.1 [80, 81], shown on the next page.

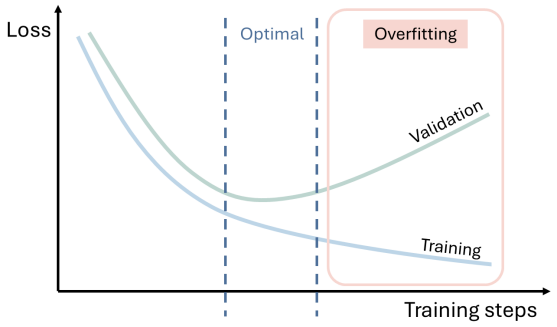


Figure 12.1: Graph showing the loss of the training set and validation set with increasing training steps. Overfitting occurs when there are too many training steps (adapted from [80, 81])

12.2. Parameter Tuning and Regularisation

Effective training of a neural network relies on the selection of hyperparameters. The values of these parameters are set prior to the training process. Commonly tuned hyperparameters include the learning rate, batch size, number of epochs as well as parameters designed to improve generalisation. Generalisation refers to the model's ability to perform well on unseen data. A well-generalised model does not overfit and does not merely memorise the training set but learns patterns applicable to new inputs.

One strategy to enhance generalisation is the application of regularisation techniques, such as L1 and L2 regularisation, dropout and early stopping. These techniques prevent the model from overfitting by introducing constraints or modifications to the training process.

In L1 and L2 regularisation, a penalty term is added to the objective function to discourage excessively large weights in the model. L1 and L2 refer to the distance metrics used to calculate the penalty. L1 regularisation adds the absolute values of the weights to the loss function, driving some weights to zero. In contrast, L2 regularisation adds a penalty proportional to the squared values of the weights. These methods are described in Equations 12.3 and 12.4, where λ is the regularisation parameter that controls the strength of the penalty. Higher values of λ enforce stricter regularisation, while lower values give more flexibility to the model.

$$\text{L1 loss function} = \text{loss function} + \lambda \sum_{i=1}^{n} |W_i| \tag{12.3}$$

$$\text{L2 loss function} = \text{loss function} + \lambda \sum_{i=1}^{n} W_i^2 \tag{12.4}$$

Dropout is another widely used regularisation technique that further improves generalisation. During training, dropout randomly deactivates a fraction of neurons in the network for each forward pass, graphically shown in Figure 12.2 [82], on the next page. This prevents the network from becoming overly reliant on specific neurons and encourages redundancy in feature learning. As a result, the model learns more robust representations and is less prone to overfitting. The probability of deactivating a neuron is controlled by a tunable parameter, referred to as the dropout rate. Typical values for this parameter range between 0.2 and 0.5, depending on the architecture and the dataset. A higher dropout rate introduces stronger regularisation but can slow down the convergence of training.

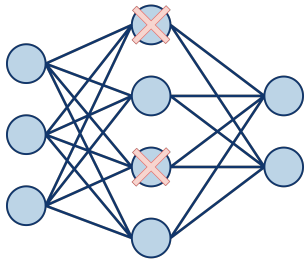


Figure 12.2: Dropout: a regularisation technique randomly deactivates some neurons (adapted from [82])

Early stopping is another effective strategy to address overfitting. During training, the model's performance on the validation set is monitored after each epoch. If the validation loss stops improving and begins to increase over consecutive epochs, training is halted. This ensures that the model achieves optimal performance before overfitting occurs.

To systematically tune hyperparameters, this work employs Optuna [83]. This is a framework for automated hyperparameter tuning. For each trial, the framework evaluates the model's performance and selects hyperparameters such as the learning rate, dropout rate and weight decay, among others, to maximise validation accuracy. This automated approach significantly accelerates the tuning process while yielding improved results.

12.3. Transfer Learning

Transfer learning is a technique where a pre-trained model, developed for a different task, is adapted to a new but related task. It leverages the knowledge encoded in the earlier layers of the model, which captures general features such as edges and textures and applies it to a specific domain. In transfer learning, the parameters of the initial layers, often referred to as the feature extraction layers, are frozen. This means that network parameters (weights and biases) in those layers remain unchanged during training, preserving their ability to extract robust, general-purpose features. Only the parameters of the later layers, which are responsible for task-specific learning, are updated. This approach reduces the amount of data and computation required for training while benefiting from the strong feature representations already learned by the pre-trained model. Transfer learning is graphically presented in Figure 12.3 [84].

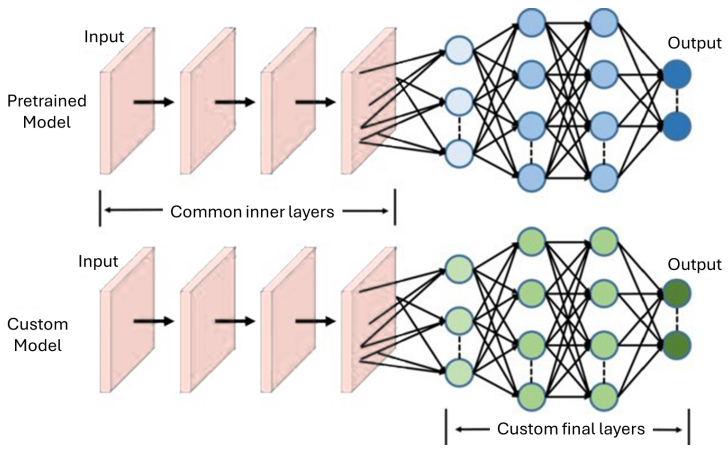


Figure 12.3: Transfer learning creates a custom model, where the parameters of the first layers remain the same and the final layers are customised (adapted from [84])

In this thesis, transfer learning was implemented using a pre-trained YOLOv8 classification model. This model, originally trained on a large and diverse dataset, was fine-tuned to classify specific print faults in 3D printing. As detailed in Section 11.3, YOLOv8's architecture is composed of 10 blocks. Transfer learning allows the flexibility to freeze a specified number of these blocks, starting from the first block (which learns low-level features like edges) to the last block (which focuses on high-level task-specific features). By freezing blocks, the model retains its pre-trained knowledge, while fine-tuning the unfrozen layers.

Moreover, transfer learning provides a scalable way to extend the model's functionality. For example, in this study, only one material in a single colour was used for training and testing. However, with transfer learning, the model can be expanded to accommodate additional materials, colours or even new types of faults with relatively little additional data. This flexibility underscores the versatility of transfer learning, making it a practical and efficient approach to adapting models to new tasks or domains. Transfer learning, however, can only be applied to domains that share significant similarities and cannot be applied to systems that serve completely different purposes.

13| Assessment of CNN Models

In this chapter, the evaluation process of the CNN models is described, including the metrics used and the datasets employed. The best-performing models are chosen based on various evaluation metrics. These metrics are calculated for the validation and test data.

13.1. Validation and Test Data

Validation data is used during the training process to monitor the model's performance after each epoch. While it does not directly influence the model's parameter updates, it acts as a measure of the model's ability to generalise to new data. The performance on the validation set can guide decisions such as early stopping, hyperparameter tuning and model selection. However, it is important to note that validation data, being part of the training process, cannot be considered completely unseen. Repeated evaluations on this data might lead to overfitting on the validation set itself. To obtain an unbiased evaluation of the model's performance, a separate test set is used. The test set remains untouched during both training and validation and is solely reserved for final performance assessment. As detailed in Section 8.2, the data was split into 70% for training, 20% for validation and 10% for testing.

Furthermore, an external test set was created to assess the model's generalisability. This external dataset comprised a new shape not included in the original dataset split (70-20-10) and was evaluated after finalising the models. The purpose of this external dataset was to evaluate the model's performance on data it has never encountered during any phase of development. Additional details about this dataset are covered in Section 8.3.

13.2. Assessment Metrics

The models are evaluated using various metrics, including accuracy, loss, confusion matrices and derived metrics such as precision, recall and F1 scores. These metrics collectively provide a comprehensive view of the model's performance across different datasets (training, validation and test).

13.2.1. Accuracy

Accuracy is one of the simplest and most commonly used metrics for evaluating classification models. It is defined as the ratio of correct predictions to the total number of predictions. The accuracy is mathematically expressed in Equation 13.1.

$$\text{Accuracy} = \frac{\text{\# Correct Predictions}}{\text{\# Total Predictions}} \tag{13.1}$$

For example, consider the case shown in Figure 13.1 on the next page, where the model correctly classifies 14 out of 16 predictions. The accuracy in this scenario is computed as $\frac{14}{16} = 87.5\%$. Accuracy can be calculated for training, validation and test datasets to observe how accurate the model's predictions are.

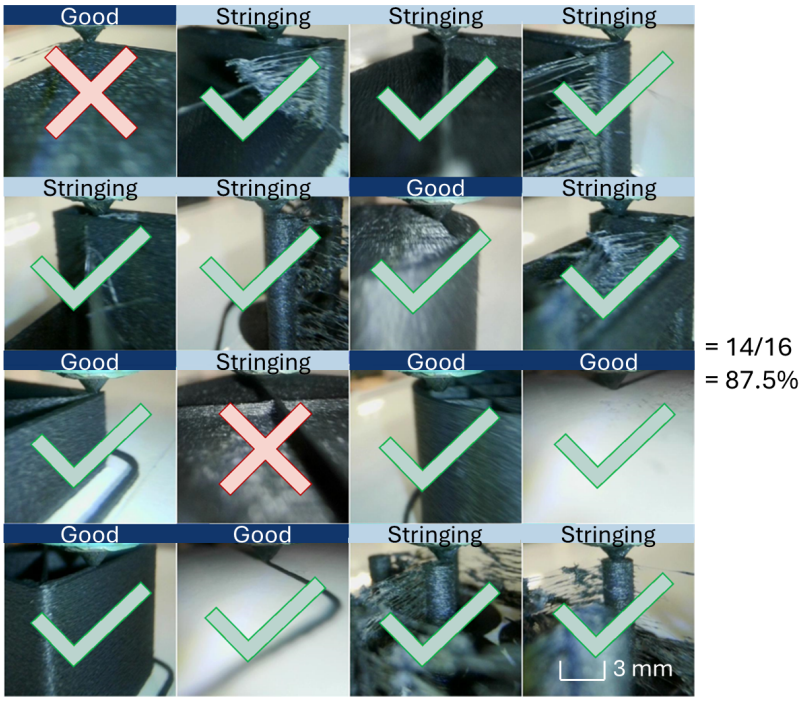


Figure 13.1: Graphical explanation of accuracy: percentage of correct classifications. In this example 14 out of 16 classifications are correct, leading to an accuracy of 87.5%

13.2.2. Loss

The loss function measures the difference between the model's predictions and the actual target values. During training, the loss is the parameter that the optimisation algorithm aims to minimise. Lower loss values indicate better model performance. In this thesis, the loss function used is the SoftMax Cross-Entropy Loss. Before calculating the loss, the raw outputs of the model are passed through the softmax function.

The softmax function normalises the raw output. The raw ouput ranges from $[-\infty, \infty]$. After applying the softmax function, these outputs are transformed into probabilities, where the sum of all probabilities equals 1. This normalisation ensures that the output represents the model's confidence in its predictions. The softmax function is defined in Equation 13.2, where z represents the raw output values, k is the number of classes and σ denotes the value after applying the softmax for each class.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{k} e^{z_k}} \qquad \textbf{for } j = 1, 2, 3, ..., k \tag{13.2}$$

Figure 13.2 illustrates the application of the softmax function, using an example with three classes. In this case, the model produces raw output values for the three classes, respectively. The output of the model is the class with the highest raw value. In this example, class three would be considered as the output or predicted class. However, these raw scores are not probabilities and lack a clear interpretation. The softmax function transforms the raw values into probabilities by first exponentiating each value (i.e., applying e^x to each), then normalising them by dividing each exponentiated value by the sum of all exponentiated values. This results in the softmax output values. These values add up to one, ensuring a valid probability distribution. The largest value, corresponding to class three, remains the highest, but the softmax output provides an interpretable confidence level for the prediction.

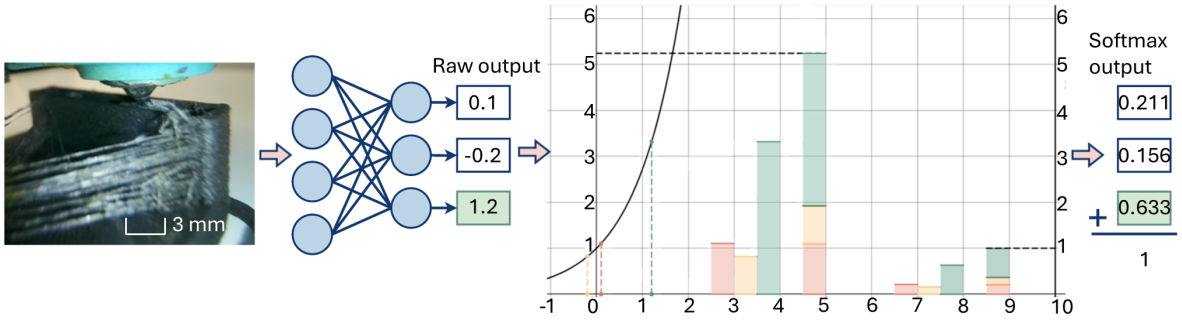


Figure 13.2: Graphical explanation of softmax function, to go from a raw output to probabilities (adapted from [85])

The cross-entropy loss is then calculated using the probabilities returned by softmax. The formula is given in Equation 13.3, where y is the output after applying the softmax and $\hat{y}$ is the target output (1 for the correct class and 0 for the other classes). Applying this to the aforementioned example yields a loss of $L = -1 * ln(0.633) = 0.457$. Note that the loss has no unit, as it is a purely mathematical construct.

$$L(\hat{y}, y) = -\sum_{j}^{k} (y_j * log(\hat{y}_j)) \tag{13.3}$$

13.2.3. Confusion Matrix

The confusion matrix is another widely used tool for evaluating classification models. It provides a detailed breakdown of the model's predictions and the actual labels. The matrix displays the true positives, false positives, true negatives and false negatives for each class. The diagonal entries of the confusion matrix represent correct predictions, while off-diagonal entries correspond to misclassifications. Figure 13.3 shows an example of a confusion matrix and its normalised version for a model distinguishing between classes good, over extrusion (OE) and under extrusion (UE). The normalised confusion matrix shows rounded proportions rather than raw counts. These values are normalised column wise. This enables easier interpretation when class distributions are imbalanced.

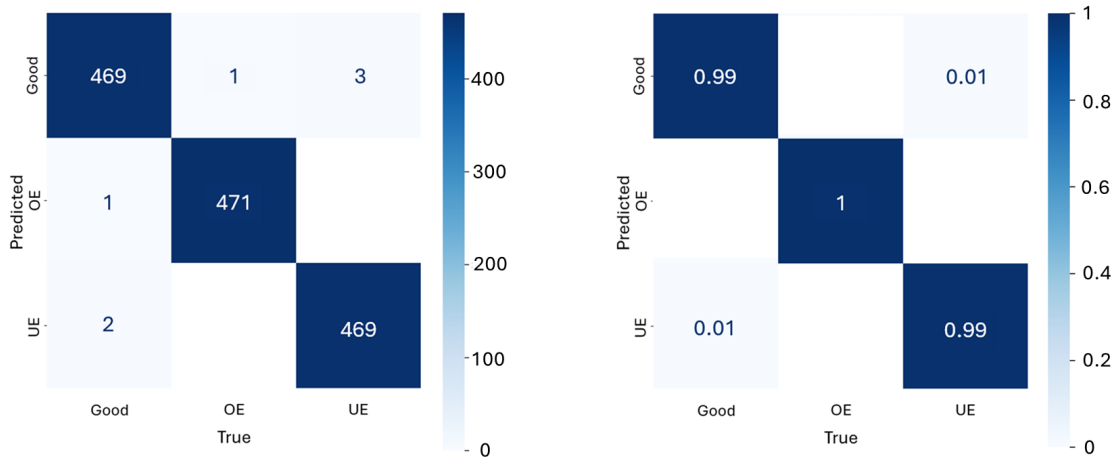


Figure 13.3: Example of confusion matrices with raw numbers (left) and normalised (right), darker blue presents a higher number

13.2.4. Precision, Recall and F1

Precision, recall and F1 scores are metrics that provide insights into the performance of a model. They are particularly useful when working with different types of errors (false positives vs. false negatives) carry varying levels of importance. Each metric is calculated per class and can also be averaged across all classes to provide a comprehensive evaluation.

Precision measures the proportion of correctly predicted positive instances out of all instances predicted as positive. It is defined for a specific class j as:

$$Precision_j = \frac{TP_j}{TP_j + FP_j} \qquad \text{for } j = 1, 2, 3, \dots, k \tag{13.4}$$

Recall, on the other hand, measures the proportion of correctly predicted positive instances out of all actual positive instances. It is defined for a specific class j as:

$$Recall_j = \frac{TP_j}{TP_j + FN_j} \qquad \text{for } j = 1, 2, 3, \dots, k \tag{13.5}$$

While precision focuses on minimising false positives, recall aims to minimise false negatives. To strike a balance between these two metrics, the F1 score is used. The F1 score is the harmonic mean of precision and recall, ensuring a single metric that considers both false positives and false negatives. It is calculated for a specific class j as:

$$F1_j = 2 * \frac{Precision_j * Recall_j}{Precision_j + Recall_j} \qquad \text{for } j = 1, 2, 3, \dots, k \tag{13.6}$$

Considering the example confusion matrix in Section 13.2.3 (Figure 13.3), the precision and recall for good, over extrusion (OE) and under extrusion (UE) are calculated and summarised in Table 13.1. The total precision recall are averaged across all classes.

Class	Precision	Recall
Good	0.9915	0.9936
OE	0.9979	0.9979
UE	0.9958	0.9936
Total	0.9951	0.9950
F1	**0.9950**	

Table 13.1: Example of precision, recall and F1

PART V

Results and Analysis

The fault detection system developed in this study involves two distinct models: one dedicated to analysing images captured from the nozzle camera and another for images obtained from the bed-side cameras. Both models underwent iterative training processes to optimise their performance and achieve the most accurate results. This part presents the outcomes derived from testing these models and provides a comprehensive analysis of the results, answering the first research subqestion posed.

Chapter 14 outlines the key observations made prior to and during the testing phase. It explains why certain errors were excluded from the models and discusses the limitations encountered in detecting specific faults. The performance metrics of the two convolutional neural network (CNN) models are reported in Chapter 15, using the evaluation methods described in Chapter 13. These metrics are further assessed against an external test set to ensure reliability and generalisation. Finally, Chapter 16 provides a critical discussion of the models' performances. The analysis compares the obtained results to initial expectations and findings from relevant literature, highlighting both the strengths and weaknesses of the detection system.

14| Printing Results

This chapter presents the observations gathered before and during the printing tests. The analysis focuses on understanding which errors the detection system can identify reliably and which errors fall outside the system's capabilities. Figure 14.1 on the next page provides an overview of the errors considered in the study, along with the reason for the exclusions. Based on these findings, the system is capable of detecting a total of 12 distinct errors. Including additional categories such as good prints, not printing and unclear images, the system operates with a total of 15 classes.

Before the testing phase started, five specific errors were excluded from the experiments: (i) small features not printing, (ii) poor surface quality at support regions, (iii) missing layers, (iv) first layer problems and (v) ripples or echos. The first four errors are not easily visible in the printed object without external references, such as CAD files or expected dimensions, to compare to the printed part. Ripples and echos require external vibration testing equipment, such as a vibration table. This was beyond the aim of the study, as a vibration table was not available. Additionally, inducing vibrations could cause unnecessary physical damage to the printer. The remaining errors were successfully printed, enabling detailed observations to be made regarding their detectability.

Certain errors were found to trigger other detectable faults rather than existing as distinct, stand-alone issues. These three errors include: support falling apart, bed separation and overheating. Support falling apart often resulted in a spaghetti failure of the support structure. Bed separation also led to spaghetti failure as the printed part detached from the bed during the process. Overheating caused multiple secondary errors, including stringing, overhang sagging, poor bridging and warping. Given their indirect nature, support falling apart, bed separation and overheating errors were not included in the final fault detection models. Instead, the focus was shifted to identifying more distinguishable errors.

Two specific faults proved to be particularly challenging for detection due to their subtle nature and limited visibility. These are: infill showing on the exterior and elephant foot. Infill showing on the exterior is only clearly visible under ideal lighting conditions. However, achieving such conditions is problematic, as the nozzle camera requires a flash light that moves with the nozzle, causing inconsistent lighting across the printed surface. Elephant foot is characterised by the slight bulging of the first printed layer. This error is subtle and often difficult to distinguish from a normal print. The limited visibility and localised nature prevented the detection system from reliably identifying this fault. To confirm these limitations, two separate models were trained: one model aimed to differentiate between good prints and the elephant foot error and another model attempted to distinguish between good prints and infill showing on the exterior. Both models exhibited accuracies below 75% during training. This validated the hypothesis that these errors are too subtle to be effectively recognised using the current detection setup.

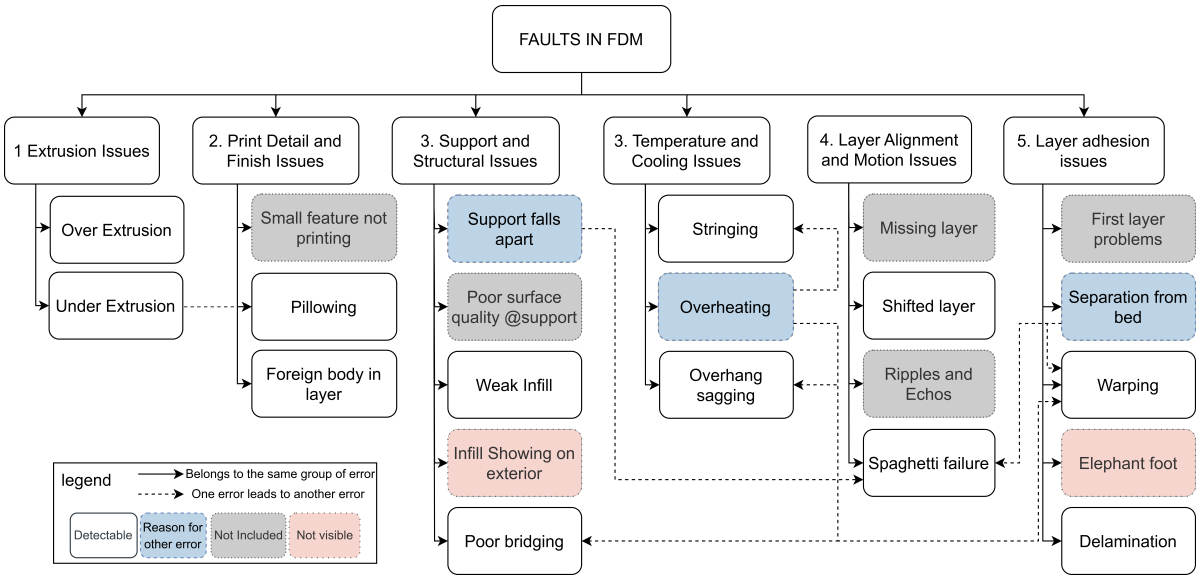


Figure 14.1: Overview of FDM errors, showing which errors the fault detection system can capture

15| Performance of the CNN Models

This chapter presents the results of the nozzle camera and the bed side cameras in Section 15.1 and 15.2, respectively, based on the metrics outlined in Chapter 13. The sections begin with the finalised tuned parameters used for training the models, detailed in Section 15.1.1 and 15.2.1. Then, the training graphs of the models are assessed in Section 15.1.2 and 15.2.2. Accuracy and loss metrics for the training, validation and test datasets are provided in Section 15.1.3 and 15.2.3. The confusion matrices for the validation datasets are presented in Section 15.1.4 and 15.2.4, while those for the training and test datasets, along with raw numerical values, are included in Appendix A. Based on the confusion matrices, precision, recall and F1 scores for the validation data are presented in Section 15.1.5 and 15.2.5. For the corresponding metrics of the training and test datasets, refer to Appendix B. Finally, Section 15.1.6 and 15.2.6 present the outcomes of testing the models with the external validation dataset.

15.1. Nozzle Camera Model

15.1.1. Parameter Tuning

Over 50 runs were performed to optimise the hyperparameters of the nozzle camera model, which are crucial in achieving optimal performance. The tuned parameters as listed in Table 15.1, were selected through experimentation and the use of Optuna, as described in Chapter 12, Section 12.2. Note that the parameters are dimensionless.

Tuned Parameter	Value [-]
Batch size	16
Learning rate	0.001476227
Dropout	0.17839
L2 regularisation	7.0953 e-06
Freeze	0

Table 15.1: Hyperparameters for the nozzle camera model

15.1.2. Training Graphs

The model was trained over 50 epochs, with performance metrics recorded at each epoch. The training and validation curves are depicted in Figure 15.1. The validation loss curve indicates effective learning of patterns, as evidenced by a sharp decline. However, from epoch 40 onwards, a rise in validation loss suggests the onset of overfitting. The best accuracy was observed at epoch 38. This model was initially selected for further evaluation.

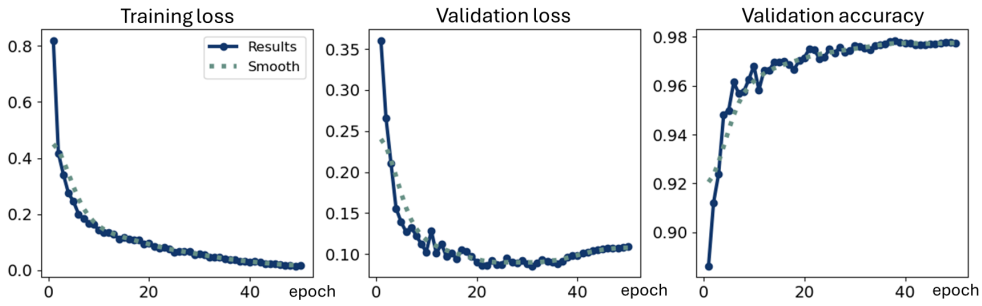


Figure 15.1: Training results of nozzle camera model: the training loss, validation loss and validation accuracy per epoch

However, after further analysing the model it could be noted that the loss of the validation was still significantly higher than the training loss, suggesting the model was overfitting. This hypothesis was confirmed using the external dataset. The model after the 38th epoch showed a low accuracy of 75.60% for the external validation. This model failed to classify over and under extrusion. The accuracies over the different epochs is plotted in Figure 15.4. Overfitting started occurring from the eleventh epoch onwards, showing low accuracies for the external validation set. Therefore, the model at the tenth epoch was preferred. This was the model where the validation and training loss were the closest and showed better performance for the external validation set. Therefore, this model was chosen as the final nozzle camera model.

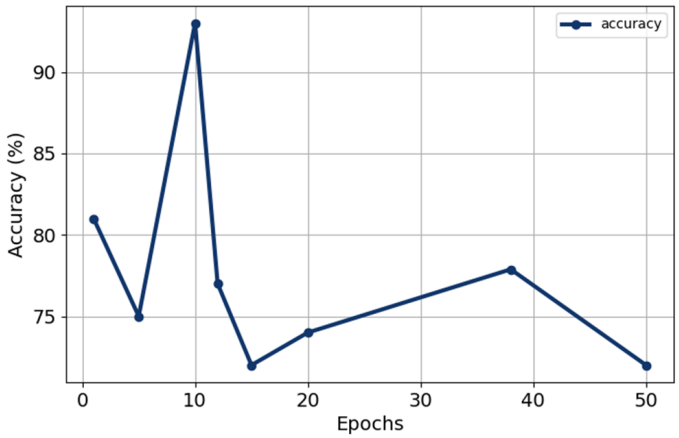


Figure 15.2: External dataset accuracies for models trained with varying epochs

15.1.3. Accuracy and Loss

The model achieved consistently high accuracy and low loss values across the training, validation and test datasets, as shown in Table 15.2. Minimal discrepancies between these three datasets indicate a well-optimised model for the nozzle camera.

Metric	Training	Validation	Test
Accuracy	99.11%	97.68%	97.09%
Loss	0.07153	0.07464	0.07478

Table 15.2: Accuracy and loss metrics for the nozzle camera model

15.1.4. Confusion Matrix

The confusion matrix for the validation dataset is shown in Figure 15.3 on the next page. This matrix represents the model's performance in distinguishing the classes. High diagonal values indicate that the model is effective at classifying the images into correct categories. The light blue boxes indicate misclassified images. Confusion matrices for the training and test datasets, along with raw numerical values, are provided in Appendix A.

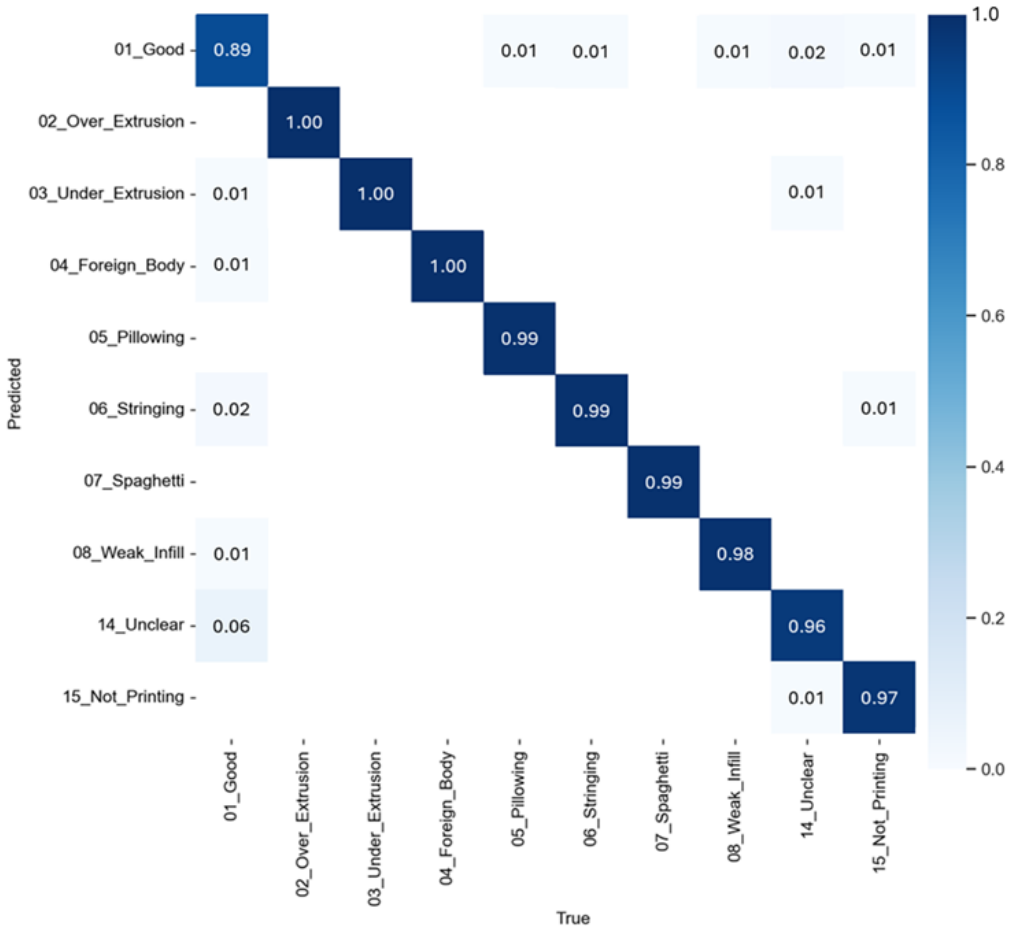


Figure 15.3: Normalised confusion matrix of the validation set of the nozzle camera model, darker blue presents a higher number

15.1.5. Precision, Recall and F1 Score

The precision, recall and F1 score provide additional insights into the model's classification performance. These metrics, summarised in Table 15.3 demonstrate the model's balanced performance across different fault categories. From the overall precision and recall, the F1 score for the nozzle camera model was calculated to be 0.9766. This reflects a high level of consistency in classifying images. Metrics for training and test datasets can be found in Appendix B.

Class	Precision	Recall
Good	0.9245	0.8891
Over Extrusion	1	0.9980
Under Extrusion	0.9841	0.9960
Foreign Body	0.9900	0.9960
Pillowing	0.9959	0.9899
Stringing	0.9742	0.9899
Spaghetti	0.9960	0.9940
Weak Infill	0.9839	0.9839
Unclear	0.9314	0.9577
Not Printing	0.9857	0.9718
Total	0.9766	0.9766
F1	**0.9766**	

Table 15.3: Precision, recall and F1 Score of validation data of nozzle camera model

15.1.6. External Validation Set

The external validation set was used to evaluate the generalisation capability of the model. The normalised confusion matrix for the external validation set is shown in Figure 15.7. The model achieved an accuracy of **93.18%**. However, the class of over extrusion showed lower performance of 76% and the weak infill class achieved a performance of 84%. The loss of the external validation set was **0.2770**, indicating an increase compared to the internal validation results. Table 15.4 summarises the precision, recall and F1 score for each class in the external validation set. From the table, it is evident that the stringing class has a significant number of false positives, while the over extrusion class show false negatives.

Class	Precision	Recall
Good	0.9638	0.9427
Over Extrusion	1	0.7642
Under Extrusion	0.6616	0.9424
Foreign Body	0.9581	0.9423
Pillowing	0.9593	0.9538
Stringing	0.7412	0.9496
Spaghetti	0.9400	0.9798
Weak Infill	0.8889	0.8421
Unclear	0.9244	0.9204
Not Printing	0.9789	0.9654
Total	0.9016	0.9203
F1	**0.9109**	

Table 15.4: Precision, recall and F1 score of external validation dataset of nozzle camera model

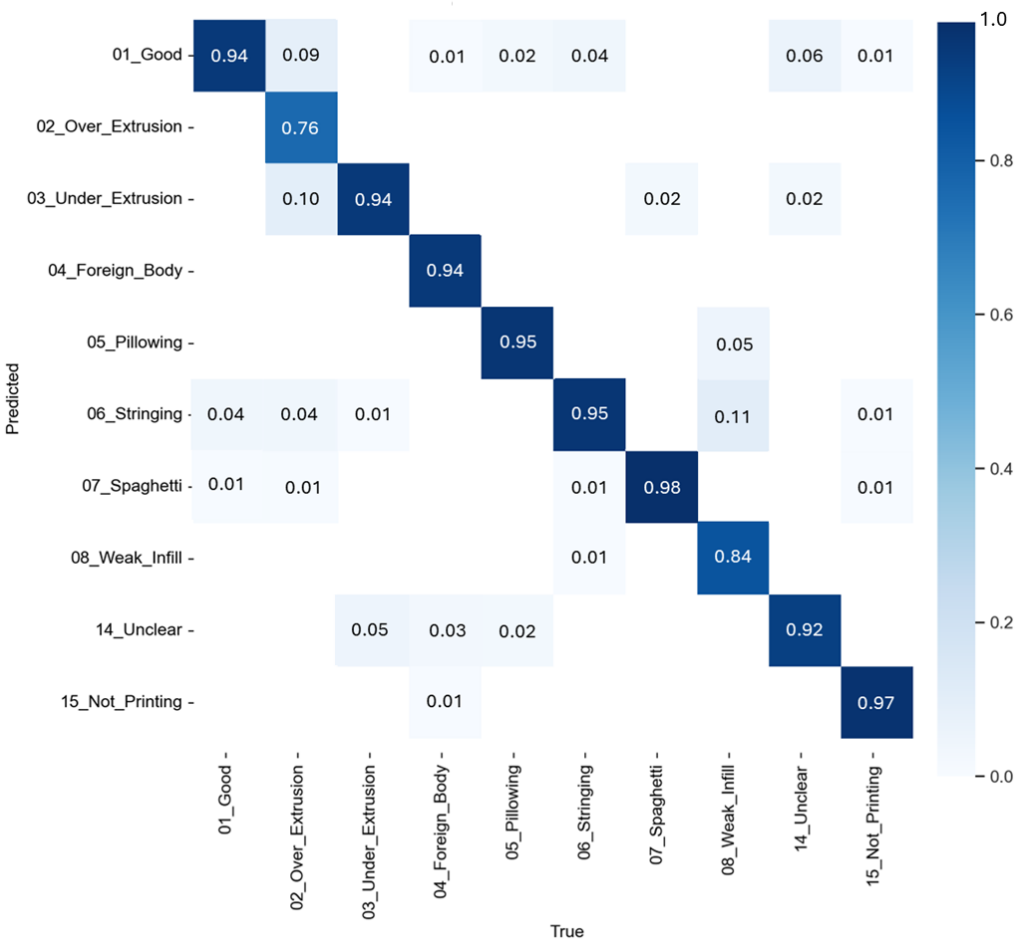


Figure 15.4: Normalised confusion matrix of the external validation set of the nozzle camera model, darker blue presents a higher number

15.2. Bed Side Camera Model

15.2.1. Parameter Tuning

The optimal hyperparameters identified for the side camera model are presented in Table 15.5. These parameters were obtained after a thorough Optuna-based tuning process.

Tuned Parameter	Value
Batch size	23
Learning rate	0.001120
Dropout	0.2288
L2 regularisation	6.891e-05
Freeze	4

Table 15.5: Hyperparameters for the side camera model

15.2.2. Training Graphs

The training process spanned 50 epochs, with the best accuracy observed at epoch 30. The model returned at epoch 30 is considered as the best side camera model. Figure 15.5 on the next page illustrates the progression of training metrics across all 50 epochs.

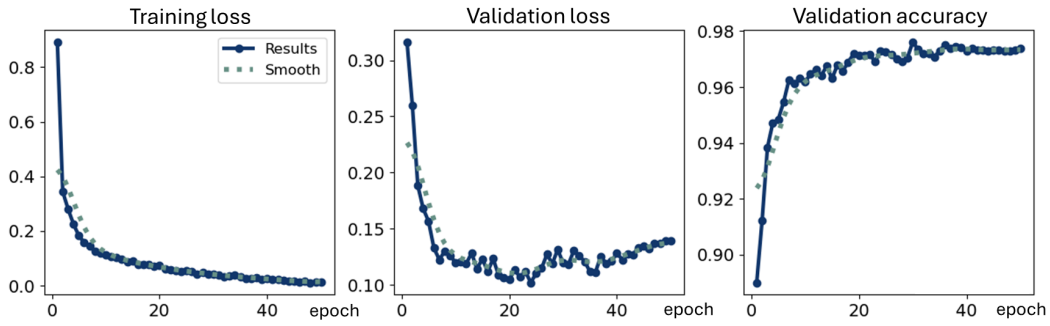


Figure 15.5: Training results of side camera model: the training loss, validation loss and validation accuracy per epoch

15.2.3. Accuracy and Loss

The side camera model demonstrated accuracies above 97% across training, validation and test datasets. The values of accuracies and losses are listed in Table 15.6.

Metric	Training	Validation	Test
Accuracy	99.81%	97.61%	97.86%
Loss	0.04255	0.1196	0.089906

Table 15.6: Accuracy and loss metrics for the side camera model

15.2.4. Confusion Matrix

The normalised confusion matrix for the validation data is shown in Figure 15.6. The confusion matrices of the training and test set, along with the confusion matrices containing raw numerical values are added to Appendix A.

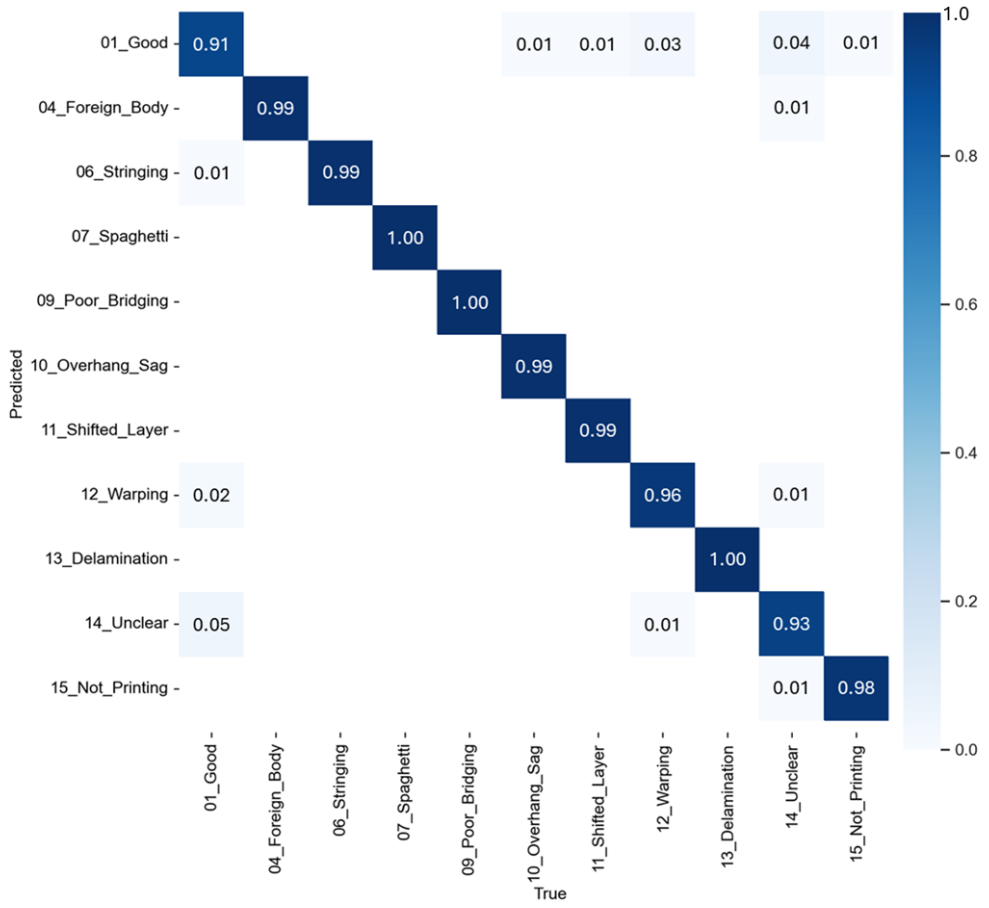


Figure 15.6: Normalised confusion matrix of the validation set of the side camera model, darker blue presents a higher number

15.2.5. Precision, Recall and F1 Score

The precision, recall and F1 score for each class are summarised in Table 15.7. The overall F1 score of 0.9764 shows high performance of the side camera model.

Class	Precision	Recall
Good	0.9019	0.9147
Foreign Body	0.9853	0.9926
Stringing	0.9900	0.9925
Spaghetti	1	0.9925
Poor Bridging	0.9975	0.9950
Overhang Sag	0.9925	0.9900
Shifted Layer	0.9975	0.9949
Warping	0.9651	0.9627
Delamination	0.9950	0.9975
Unclear	0.9328	0.9282
Not Printing	0.9851	0.9778
Total	0.9766	0.9762
F1	**0.9764**	

Table 15.7: Precision and recall of validation data of side camera model

15.2.6. External Validation Set

The external validation set was used to evaluate the generalisation capability of the side camera model. Figure 15.7 depicts the normalised confusion matrix of the external set. The diagram highlights offsets between performances of different classes. It can be noted that there is low performance in classifying specific errors: stringing, poor bridging and shifted layer. The model confuses these classes with warping and overhang sag. The model has an overall accuracy of **89.3%**. The loss for the external validation set was **0.3927**, indicating an increase compared to the internal validation results.

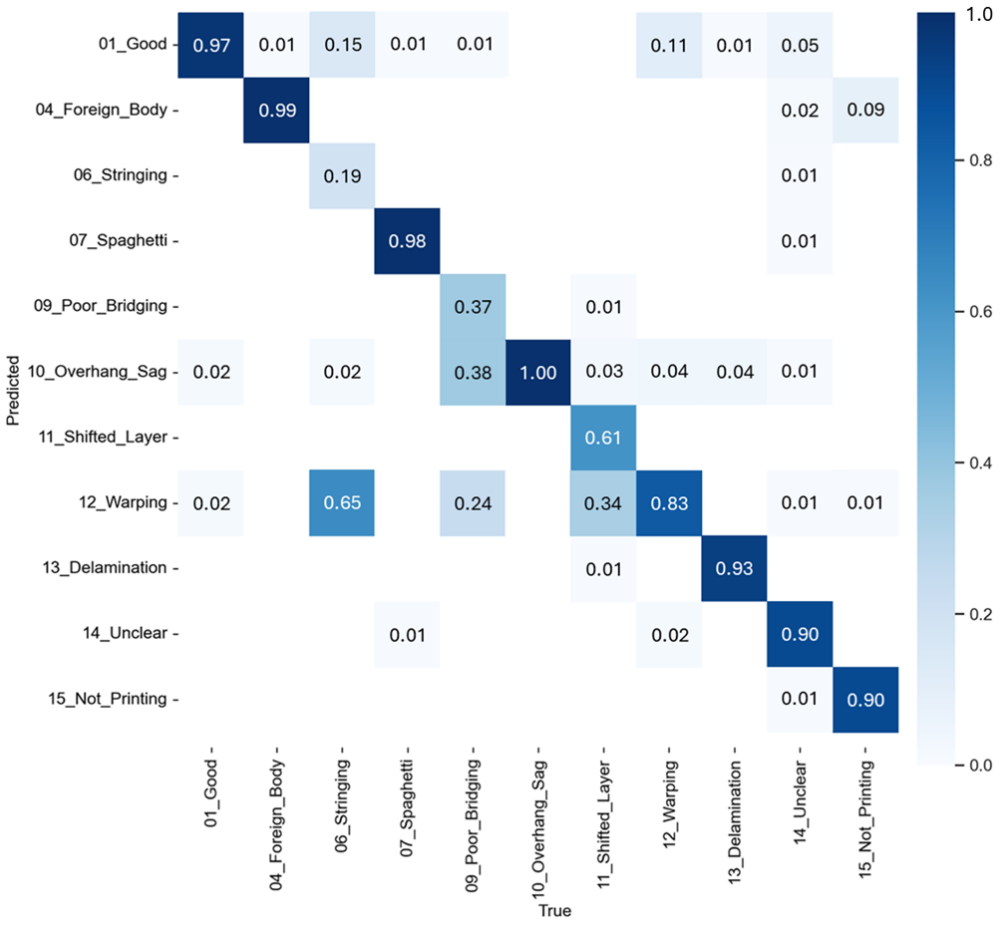


Figure 15.7: Normalised confusion matrix of the external validation set of the side camera model, darker blue presents a higher number

The trends in precision and recall are summarised in Table 15.8. For the classes stringing, poor bridging and shifted layer, precision remained high. However, their recall values were notably low, highlighting challenges in detecting these specific errors. Conversely, classes such as warping and overhang sag showed the opposite trend, with lower precision but higher recall.

Class	Precision	Recall
Good	0.8575	0.9698
Foreign Body	0.9748	0.9891
Stringing	0.8000	0.1935
Spaghetti	0.9948	0.9793
Poor Bridging	0.9750	0.3714
Overhang Sag	0.7433	0.9955
Shifted Layer	0.9891	0.6149
Warping	0.6220	0.8327
Delamination	0.9964	0.9349
Unclear	0.9703	0.8969
Not Printing	0.9684	0.9020
Total	0.8992	0.7891
F1	**0.8406**	

Table 15.8: Precision, recall and F1 of external validation dataset of side camera model

16| Discussion of the Models' Performance

This chapter discusses the results presented in Chapter 15. In Section 16.1, the interpretations of the results are discussed. In Section 16.2, the performance of the models are compared to what is known in literature. The strengths and weaknesses of the models are scrutinised in Section 16.3.

16.1. Interpretation of the Results

16.1.1. Nozzle Camera Model

The results of the nozzle camera model demonstrate high overall performance and generalisability, underscoring the effectiveness of the chosen hyperparameters and training strategy. The model consistently achieved high accuracy, with a peak validation accuracy of 97.68% and a comparable test accuracy of 97.09%. This indicates minimal overfitting to the training data. Furthermore, the low training, validation and test losses signify that the model effectively captured the underlying data patterns with minimal prediction errors. An analysis of the confusion matrix for the validation and external datasets reveals the model's robustness in classifying most fault categories with high precision and recall.

However, a few challenges were observed in certain categories, particularly in the external validation dataset. For instance, the over extrusion class showed a noticeable drop in recall, indicating a higher rate of false negatives. This suggests that the model struggles to identify over extrusion faults in certain scenarios, possibly due to feature overlap with other classes or that there is insufficient diversity in the training samples for this defect. A significant amount of images, namely 19%, were classified as good or under extrusion instead. The stringing class exhibited a lower precision, indicative of a higher rate of false positives. This could point to misclassifications with visually similar defects, such as weak infill. The class weak infill showed lower performance, However, no significant conclusions can be drawn from this result as the weak infill class of the external validation set only contained 19 images.

The external validation accuracy of 93.18% demonstrates the model's ability to generalise to unseen data, even when the images were sourced from a separate dataset. The increased external validation loss, in comparison to the internal validation set, reflects the domain-specific challenges. These challenges are due to specific variations, such as lighting or different angles. This increase in loss highlights the complexity and the difference of the new data rather than indicating overfitting, as the model continues to have an accuracy above 90%.

The F1 score for the validation set and the external validation set reflect a well-balanced performance across precision and recall metrics. The minor decline in the external F1 score indicates that the model's performance is only slightly impacted by unseen variations, reinforcing its robustness.

16.1.2. Side Camera Model

The results of the side camera model highlight its strong overall performance in classifying various printing errors and conditions, with high accuracies observed across training, validation and test datasets. Despite these positive outcomes, specific challenges emerged, particularly with the external validation dataset.

The internal validation metrics showed consistent accuracy and relatively low loss values. This indicates that the model learns effectively from the training data. The confusion matrix (Figure 15.6) for the validation dataset demonstrates robust performance, with minimal misclassifications across the 11 classes. The overall F1 score of **0.9764** reinforces the reliability of the model within the internal dataset. These findings are further supported by the precision and recall values. They indicate that the model effectively balances between correctly identifying a class (recall) and avoiding false positives (precision).

However, when tested with the external validation dataset, the model's accuracy dropped to **89.3%** and the loss increased to **0.3927**. These changes suggest challenges to analyse new data, particularly for classes such as stringing, poor bridging and shifted layer. As detailed in the confusion matrix, significant interference exists between these classes and warping. This observation is verified by the relatively high precision but low recall values for stringing and poor bridging. For warping, the opposite trend is observed, with lower precision and higher recall. However, upon closer inspection of the misclassified images, indications of warping could be found in several images from the stringing and shifted layer classes. This suggests that multiple visible errors co-occurred in these cases, indicating that the model's actual performance is better than what is reflected by the confusion matrix.

Additionally, the precision for the overhang sag class was observed to be relatively low, despite its high recall. This indicates an increased amount of false positives. One hypothesis for this sub-optimal behaviour is that the confusion is possibly attributed to the shape of the external test print. The test prints contains particularly dominant overhanging features. Inducing errors, using high-temperature settings, could inadvertently cause overlapping visual features of overhang sag. This possibly caused this particular confusion.

Overall, the side camera model demonstrates robust classification capabilities for the majority of classes, even in the presence of challenging external datasets. However, the discrepancies in certain classes highlight areas for further improvement, such as addressing overlapping error features and enhancing the model's ability to generalise to even more diverse test conditions. Incorporating additional training data with better representation of these error types and refining the model architecture could improve the model's robustness and adaptability to unseen datasets.

16.2. Comparison to Literature

The performance of the models developed in this study is compared to relevant literature to assess the effectiveness and contextualise the results within the field of fault detection in FDM 3D printing. Most studies used internal validation to assess their results, which may limit the generalisability of the published findings. The main improvement of our model compared to others is that it handles more error types, while achieving similar or higher accuracies.

The nozzle camera model achieved a validation accuracy of 97.68%, with robust generalisation to unseen data as indicated by an external validation accuracy of 93.18%. These results align closely with or outperform comparable studies that also employed a nozzle camera in FDM desktop printers. For example, Jin et al. [32] reported a test accuracy of 91.0% for the classification of the height of the nozzle, using white PLA printing with a Prusa MK3. Another study performed by Jin et al. [34] achieved 98% accuracy for the classification between good, over and under extrusion. Moreover, the authors used this model to successfully create an in-situ correction system, using white PLA on a Prusa MK3. Brion and Pattinson [26] made an error detection and correction system that predicts the printing parameters: flow rate, speed, temperature and Z-offset and classifies them between low, good and high. Their research achieved an overall validation accuracy of 90.2%, after applying transfer learning twice. They printed in PLA using varying colours using a Creality 20 printer. These studies were used to set a minimum validation accuracy as benchmark. This minimum accuracy was set to 91%. This was chosen as it is the lower end of achieved accuracies for a comparable nozzle camera models in other research. Although our model contained more classes to classify and utilised a different material in a challenging colour (black), it achieved accuracies similar to those found in literature.

The side camera model also displayed strong performance, with an internal validation accuracy 97.61% and an external validation accuracy of 89.3%. When compared to existing literature that made CNN models for bed side cameras, its performance is competitive. For instance, Rettenberger et al. [2] reported a test accuracy of 83% for classifying faults stringing, under extrusion and good. In their research, white PLA was printed using the Creality Ender-3Pro. Karna et al. [86] achieved 89.7% accuracy for detecting errors including stringy, obstacle and unstick defects. They used two printers for their research: a Creality Ender 5 and a Prusa MK3s+. The study by Brion et al. [35] designed a warping detection system, achieving an accuracy of 88.7%. They utilised different colours of ABS on a Creality CR-20. Similar studies achieved accuracies of 99.3% [36] and managed to create a warping correction system, using while PLA on a Prusa MK2s. Kim et al. [87] created a spaghetti detection system with an accuracy of 96%, using varying colours. Though this literature utilised different materials, colours and different printers, they can be used as a benchmark. From these results, a minimum validation accuracy as set to 83%. Our side camera model contains more error classes and does not show lower performance than similar models found in literature.

Other fault detection systems leveraging CNNs for FDM, such as [88], which utilised a top view camera achieved 84.95% validation accuracy for binary classification. Wang et al. [89] utilised a rotating bed camera and achieved 91% accuracy for five surface defects. They demonstrated lower performance

when compared to the models developed in this study. Additionally, Okaro et al. [90] leveraged a CNN model for laser powder bed fusion (L-PBF), using photodiodes, achieving an overall accuracy of 77%. Bhattacharya et al. [91] on the other hand leveraged NN for error detection in vat polymerisation, with a resulting accuracy of 80%. While these studies achieved excellent results for specific defects, the nozzle and side camera models in this thesis are distinguished by their versatility in addressing a wider range of fault types, while keeping high accuracies.

In summary, the models developed in this study outperform or at least match state-of-the-art systems in terms of accuracy. Our models especially outperform existing research in the context of detecting a diverse range of printing errors. These findings position the developed models as a significant contribution to real-time fault detection in FDM 3D printing. Table 16.1 provides a concise overview of similar research that used NN, which sensor they used, the classes they classify and their achieved accuracies. This comparison is further visualised in Figure 16.1, where the amount of error types is plotted against the validation accuracies of existing literature. The results underscore the robustness of the models while highlighting areas for future work, such as improving performance for overlapping defect types.

Source	Printer	Sensor	Classes	Acc. (%)
[32]	FDM	nozzle camera	nozzle height: high +, high, good, low	91.0
[34]	FDM	nozzle camera	good, over, under extrusion	98.0
[26]	FDM	nozzle camera	flow rate, speed, temp, Z-offset: low, good, high	90.2
[2]	FDM	bed side camera	normal, stringing, under extrusion	83
[35]	FDM	bed side camera	warping detection	88.72
[36]	FDM	bed side camera	warping detection	99.3
[86]	FDM	bed side camera	stringy, obstacle, unstick	89.7
[92]	FDM	bed side camera	acceptable, faulty	70.0
[93]	FDM	bed side camera	Print speed value, temperature value	94.0
[87]	FDM	bed side camera	spaghetti object detection	96.0
[89]	FDM	Rotating bed camera	normal, blob, void, thick line, crack, misalignment	91.0
[88]	FDM	top-view camera	acceptable, faulty	84.95
[91]	VAT	temperature, vibration, humidity	acceptable, faulty	80
[90]	L-PBF	photodiode	acceptable, faulty	77
[94]	L-PBF	thermographic camera	ok, metal splatters and delamination	96.8
own	FDM	nozzle and 2 bed cameras	total of 15 classes , containing 13 errors	97.7, 97.6

Table 16.1: Comparison of fault detection methods in additive manufacturing

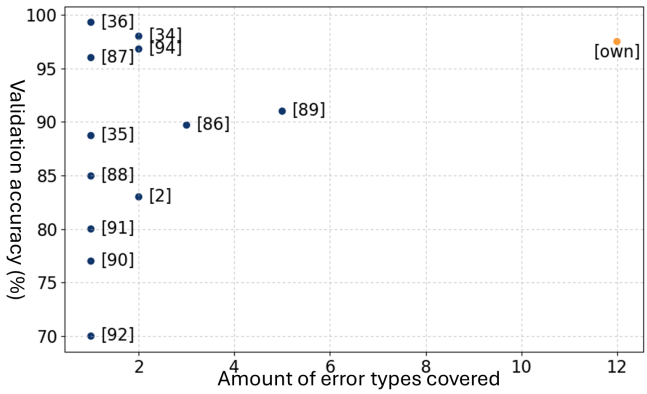


Figure 16.1: Validation accuracy of our models compared to existing literature, considering the amount of error types covered

16.3. Strengths and Weaknesses of the Models

16.3.1. Nozzle Camera Model

The nozzle camera model demonstrates notable strengths, including its high generalisability and excellent overall performance. With a validation accuracy of 97.68% and an external validation accuracy of 93.18%, the model has proven to be both robust and effective in making reliable predictions. Its low training, validation and test losses reflect minimal prediction errors. Additionally, the model excels at detecting a wide range of faults, as evidenced by its high precision and recall of the internal dataset for most defect categories. The balanced performance metrics, particularly the high F1 scores, ensure accurate classification without bias toward any specific classes. When compared to similar studies, the nozzle camera model either matches or exceeds the performance of comparable systems, showcasing its strong design and implementation.

However, our nozzle camera model has some weaknesses, particularly in detecting certain fault classes. It struggles with over extrusion, as evidenced by a lower recall of the external validation set of 76.42%. This indicates difficulty in distinguishing this defect from others due to overlapping visual features. Similarly, the precision for the stringing class of the external data (74.12%) highlights a tendency toward false positives. This is likely caused by confusion with visually similar defects such as weak infill. These weaknesses should be addressed by creating more diverse data.

16.3.2. Side Camera Model

The side camera model showcases several strengths. It achieved a strong internal performance, with a high F1 score of 0.9764 and minimal misclassifications across its 11 classes in the internal dataset. Its ability to balance precision and recall for most fault categories highlights its reliability in identifying defects while minimising false positives. The model effectively detects a broad range of 3D printing issues from the side camera perspective. In external validation, the model achieved a competitive accuracy of 89.3%, indicating its ability to generalise to unseen data despite variations in defect types.

Similar to the nozzle camera model, the side camera model exhibits some weaknesses, particularly in its performance on external datasets. While the external validation accuracy of 89.3% is competitive, the increased loss (0.3927) highlights difficulties in adapting to domain shifts, such as changes in lighting or print characteristics. This suggests the model may struggle when exposed to conditions that deviate from its training data. This weakness can be solved by including a more diversified dataset, increasing the generalisability of the model.

PART VI

Fault Detection System

With models achieving a validation accuracy of over 97% and demonstrating sufficient performance on an external validation set, their integration into a comprehensive fault detection system becomes feasible. This system leverages the outputs of three cameras: the nozzle camera and two bed cameras. Each camera plays a distinct role, as certain faults are only detectable from specific perspectives. The fault detection system encompasses multiple interconnected steps: importing and preprocessing images, analysing them with trained models and combining their outputs using a defined logic to determine whether a fault is present.

This part aims to describe the designed fault detection system, based on the error logic. This forms the basis for answering the second subquestion. In Chapter 17, the detection system is examined, providing an in-depth explanation of its various components and operational steps. Chapter 18 outlines the error logic employed in the fault detection system, detailing the conditions under which faults are identified. Finally, in Chapter 19 the strengths and weaknesses of the system are discussed.

17| Outline of the Detection System

The detection system is composed of several interconnected components and operates in a continuous loop. An overview of the process is provided in Figure 17.1. In each iteration of the loop, three images are captured: one from the nozzle camera and two from the side cameras (left and right). These images are then processed differently based on their source, leading to three outputs that are subsequently integrated using the logic described in Chapter 18.

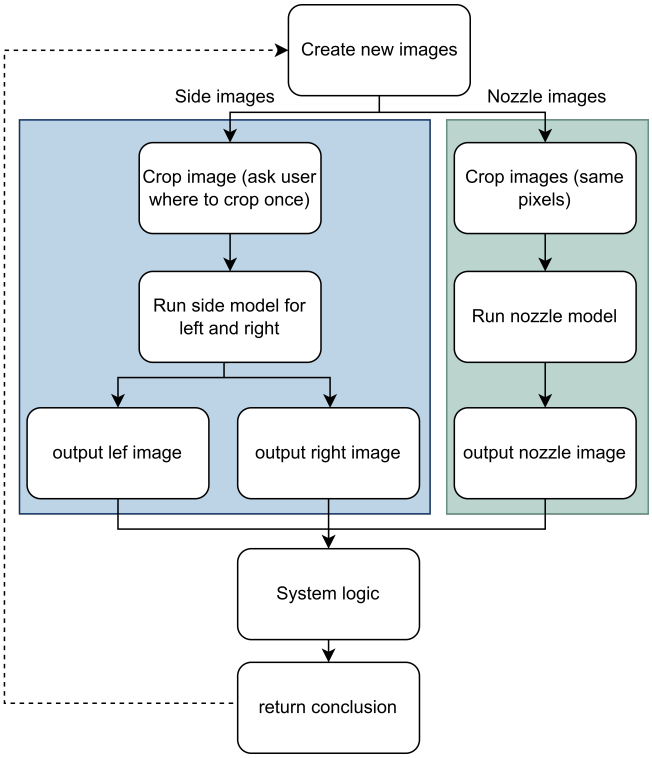
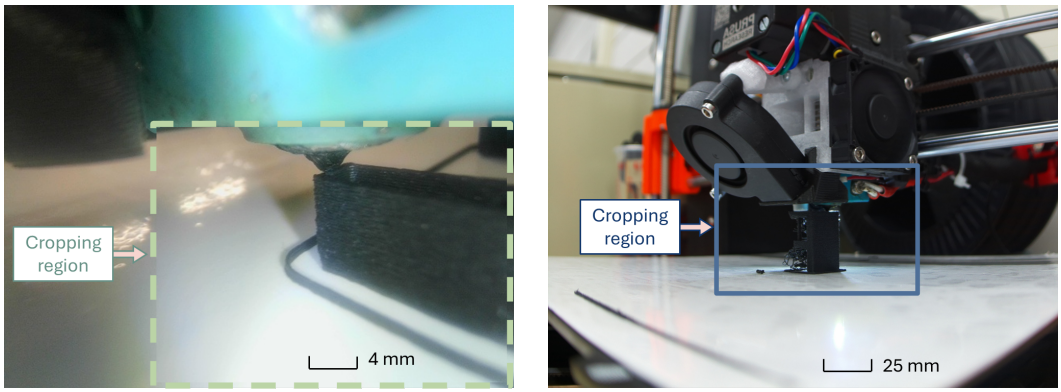


Figure 17.1: Overview of the looping detection system: first images are created, then these are processed based on their camera before the results are combined into one conclusion

The image processing workflow varies for the nozzle camera and the side cameras due to differences in their perspectives and the movement of the print bed. Cropping is applied to all images to reduce the computational load on the models.

For the nozzle camera, cropping is consistent across all iterations of the detection system. The nozzle does not move relative to the camera, ensuring a fixed cropping region. Each image is cropped from 640x480 to 440x320 pixels, as shown in Figure 17.2a on the next page. This fixed cropping simplifies the process and ensures that all nozzle camera images are uniformly preprocessed.

For the side cameras, cropping requires manual adjustment during the first loop. This is because new prints can be positioned differently on the bed, resulting in no fixed reference point for the camera. To address this, the user is prompted to select the cropping field during the first iteration. The cropping is achieved by moving a selection box of 1600x1000 pixels over the image of 4056x3040 pixels and pressing 'c' to confirm. This process ensures that the cropped area accurately captures the print region for subsequent loops. Figure 17.2b on the next page shows an example of the cropping process of the side images.



(a) Nozzle camera: cropping to 440x320 pixels, where the cropping box is fixed

(b) Side camera: cropping to 1600x1000 pixels, where the cropping box can move

Figure 17.2: Cropping processes

Once the images are cropped, they are processed using their respective Convolutional Neural Network (CNN) models. The nozzle camera images are analysed using a model trained specifically to detect errors observable from the nozzle's perspective. Similarly, the side camera images are processed using a separate model tailored to identify errors detectable from the side views. These models are described in detail in Part V of this report.

The outputs from all three models are then passed to the logic system for integration, which combines the results into a unified conclusion. This conclusion reflects all detected errors and is returned to the user. The integration process ensures that errors are appropriately prioritised and accurately reported.

18| Logic of the Detection System

In the fault detection system, there is a distinction between the raw output generated by the models and the final errors reported by the system. The model outputs represent the predictions before any logic is applied, while the returned errors are determined after a structured decision-making process. This logic to come to the final returned error consists of two steps. First, the **model outputs** are processed into **individual answers** for each camera. Secondly, these answers are combined into a **single conclusion** for the entire system. The first step is detailed in Section 18.1, which describes how repeated outputs are used to ensure consistency before returning an individual answer. The second step, which integrates the answers into one cohesive conclusion, is discussed in Section 18.2.

18.1. Returning Individual Answers per Camera

Deriving answers from the raw model outputs for each camera is illustrated in Figure 18.1 on the next page. It outlines the system logic for handling individual camera outputs and includes an example. The logic begins with feeding in the last predictions provided by the models. There are three separate lists that store the outputs of the three cameras. In the provided example the last outputs are 'good,' 'foreign,' and 'foreign' for the nozzle, left and right cameras, respectively. To avoid interference from previous iterations, the system initialises the answers as 'none' at the start of each loop. These initial 'none' values are subsequently updated under specific conditions: when the output is 'good', 'unclear' or an error is consistently repeated.

18.1.1. Occurence of Good, Unclear or Not Printing

When the output is classified as good, unclear or not printing, special handling is applied to ensure consistency in the final answers. In all three cases, that last output is removed from the list saving all previous outputs for that camera. This is done to maintain the continuity of consecutive answers. For example, if the output contains 20 repetitions of stringing, followed by one instance of unclear and then stringing again, the system will still correctly identify stringing as the last answer.

If the model's output is good, the camera's answer changes to 'good'. Additionally, the system keeps track of the number of consecutive good outputs. Once this count reaches 4 or more, the entire list is cleared to prevent rare erroneous outputs from affecting the conclusion. For instance, if there are 1500 predictions in total, with 1496 classified as good and only 4 scattered outputs as stringing, the system avoids incorrectly returning stringing. Similarly, when the output is unclear, it is included in the list and then removed to prevent disruption of consecutive error tracking. In the case of not printing, the output is excluded from the list entirely unless at least two cameras simultaneously detect this state. This approach accounts for scenarios where the nozzle camera may predict not printing during travel movements, which do not represent an actual fault.

18.1.2. Consecutive Answers

The number of times a specific output must be repeated before being confirmed as an error is determined through statistical analysis. This threshold is chosen to ensure a detection system with a minimum accuracy of 99%, while the model's accuracy is only 97%. If one camera consistently returns

the same error for 4 iterations, the error is confirmed. Notably, one intermittent occurrence of a good state is allowed. For example, if the predictions are [stringing, stringing, good, stringing, stringing], stringing is returned as the confirmed error.

The chance of more than one wrong answer occurring in a list where at least 4 answers equal the correct answer is calculated using the formula presented in Equation 18.1. Here n is the number of trials in total, p the probability of a success (97%) and k the number of successes. This could only occur when there are 2 errors in a list of 6 or 3 errors in a list of 7, otherwise the count of good reaches 4 and the list is deleted. The chances of these occurrences are $P(k \leq 2, n = 6) = 0.001157\%$ and $P(k \leq 3, n = 7) = 0.002636\%$ respectively, which are negligible.

$$P(k) = \binom{n}{k} p^k (1-p)^{n-k} \tag{18.1}$$

Using four consecutive answers and one other prediction, the chance of the largest class being correct is: 99.99%. The latter is found using Equation 18.2. This equation applies Equation 18.1 in a binary classification scenario where there are only two possible outcomes (e.g., stringing or good).

$$P(\text{largest class is correct}) = \frac{P(k=4)}{P(k=4) + P(k=1)}, \text{ with } n = 5 \tag{18.2}$$

Additionally, if two cameras detect the same error for 2 instances, the answers become that error. In case all three cameras detect the same error simultaneously, the three answers equal this error.

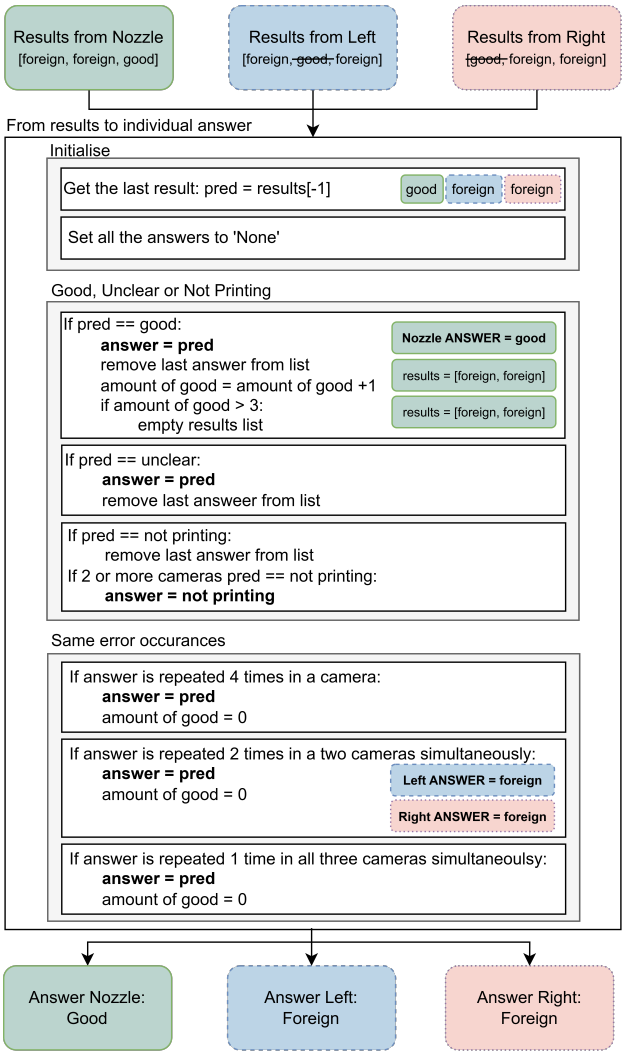


Figure 18.1: From model outputs to answers, a logic for good, unclear and not printing occurrences and ensuring consecutive answers

18.2. Integration of Camera Outputs

Once individual predictions from each camera are obtained, they are passed to the next function for integration into a final conclusion. This process is illustrated in Figure 18.2 on the next page. The integration system prioritises errors based on their detectability by specific cameras.

Four errors are detectable exclusively by the side cameras: overhang sag, poor bridging, warping and delamination. The nozzle camera images taken during those errors were passed through the nozzle camera model, to see the effect of an error visible by the side cameras only. This analysis revealed that the nozzle camera misclassified around 50% of the images when overhang sag or delamination occurred. The outputs were often misclassified as extrusion errors, stringing or pillowing. In the case of warping, no significant misclassification trends were observed. Therefore, when overhang sag, poor bridging or delamination are detected by the side cameras, the predictions from the nozzle camera are disregarded.

On the other hand, only the nozzle camera is capable of detecting these four errors: over extrusion, under extrusion, pillowing and weak infill. Among these, over extrusion and under extrusion occasionally resulted in misclassifications by the side cameras, such as warping, unclear image classifications. These misclassifications were associated with high loss values, indicating significant confusion. As a result, the side cameras are excluded from consideration when over extrusion or under extrusion is detected by the nozzle camera.

Importantly, the side camera detection of overhang sag, poor bridging and delamination takes priority over nozzle camera predictions of extrusion-related errors. This decision was made because extrusion errors were frequently reported by the nozzle camera in instances where the side cameras detected overhang sag or delamination. In contrast, when actual over extrusion or under extrusion occurred, the side cameras did not classify these as overhang sag, poor bridging or delamination.

Finally, the error logic system integrates all predictions to generate a single conclusion. In case all three answers are the same, the conclusion is defined as that given answer. In case all three are classified as either good, unclear or none, the conclusion is made to be good. In cases where multiple errors are detected simultaneously, the system reports all identified errors as part of the final output.

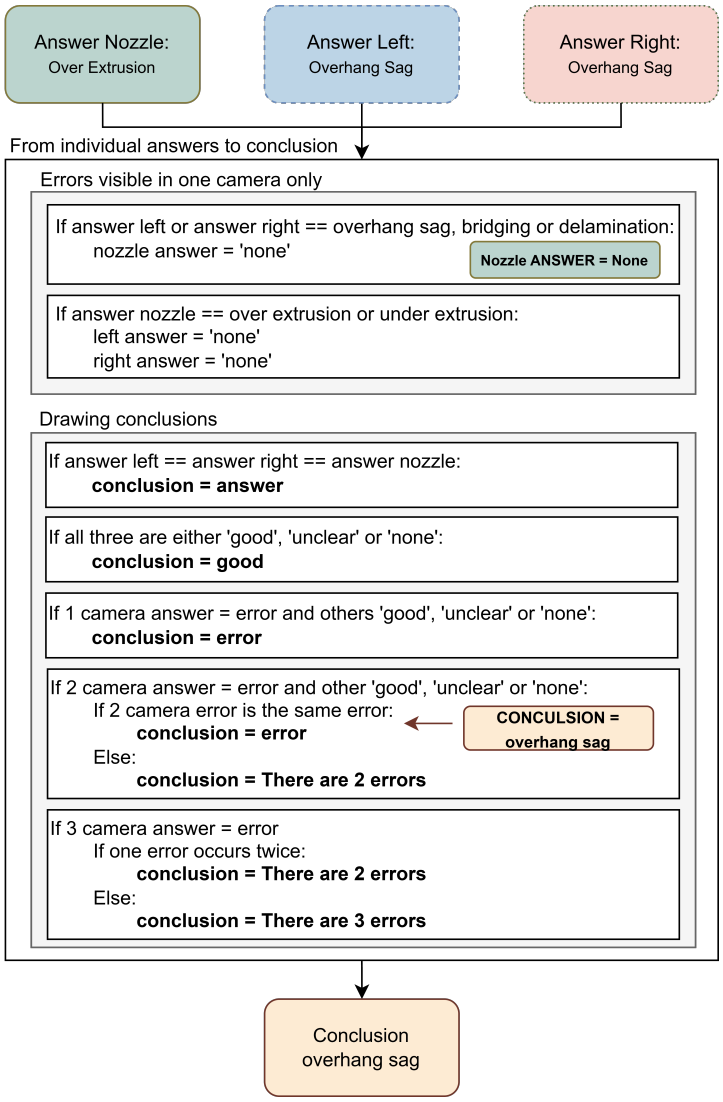


Figure 18.2: System logic: from individual answers to a final conclusion, including a prioritising system

19| Strengths and Weaknesses of the Fault Detection System

The developed fault detection system provides a structured and systematic approach to identifying errors in FDM printing. By integrating multiple viewpoints and leveraging highly accurate machine learning models, the system successfully detects a wide range of common print failures. However, while the system performs well within its training conditions, certain limitations exist in its ability to generalise. This chapter discusses the strengths and weaknesses of the system, providing insights into its current capabilities and areas for improvement.

19.1. Strengths of the Fault Detection System

The fault detection system presents several key strengths that contribute to its effectiveness:

- **Comprehensive error coverage:** The system is capable of detecting 12 distinct error types, ensuring that a broad spectrum of common 3D printing failures is addressed.
- **Multi-viewpoint integration:** Three separate viewpoints (the nozzle camera and two bed cameras) are systematically combined using a priority-based decision system, allowing for a more robust and accurate error detection process.
- **High model accuracy:** The classification models used in the system achieve validation accuracy exceeding 97.5%, indicating strong performance within the trained dataset.
- **External validation:** The system has undergone external validation, demonstrating its ability to function beyond the initial training dataset, reinforcing confidence in its real-world applicability.

19.2. Weaknesses of the Fault Detection System

Despite its strengths, the fault detection system has certain limitations that need to be addressed:

- **Excluded errors:** While 12 error types are included, 7 additional errors remain undetectable due to insufficient visibility or the requirement of an external reference for identification.
- **Reduced performance on external data:** Although externally validated, the system exhibits lower detection accuracy on unseen dataset.
- **Shape-specific validation:** The external validation was performed on a single shape.
- **Material and colour constraints:** The models were trained exclusively on black filament using a single material, limiting their applicability to prints with different colours and material properties.

To reduce the limitations of the fault detection system, several improvements can be considered. Expanding the dataset to include multiple filament colours and material types would improve the system's generalisation capabilities, ensuring reliable performance across a broader range of prints. Additionally, validating the system on multiple geometries would help assess its adaptability to different shapes. Finally, exploring methods to incorporate additional error types, particularly those that currently require external references, could further enhance the system's effectiveness in real-world applications.

PART VII

Exploring Future Applications, Opportunities, and Their Impact

To take this detection system further, an integrated fault detection and correction system can be designed. This would enhance the system's impact, ultimately making the 3D printing process closed-loop. Moreover, the fault detection system is primarily designed for the Prusa Mk3s, but can be extended to other similar desktop 3D printers. With some adaptations, the system can also be applied to variations such as 45° belt printing and large-scale printing. Large-scale printing frequently incorporates 45° printing techniques [95].

This part explores the potential for a fault correction system. Moreover, this part discusses applying the fault detection to the aforementioned advanced printing methods, as well as the implications and adjustments required for their integration. Chapter 20 discusses the possibilities for creating an integrated correction system. Chapter 21 focuses on 45° belt printing, while Chapter 22 addresses large-scale 3D printing. In addition, the broader impact of fault detection systems is quantified, considering both small-scale and large-scale printing applications. In Chapter 23, the potential of the integrated detection system in large-scale printing is illustrated, by application of data from Tanaruz [1]. Through these discussions, this part addresses the third and fourth subquestions of the thesis, posed in Chapter 2. Finally, in Chapter 24, the significance of this work and its findings is summarised.

20| Possibilities for Correction System

The system developed in this work represents the first step toward creating a fully integrated fault detection and correction system for 3D printing. A conceptual overview of the combined fault detection and correction system is shown in Figure 20.1. This chapter explores the steps required to achieve this and discusses correction mechanisms customised to different error types. The future steps required to transition from detection to correction are discussed in Section 20.1, while Section 20.2 examines specific correction mechanisms for various detectable error types.

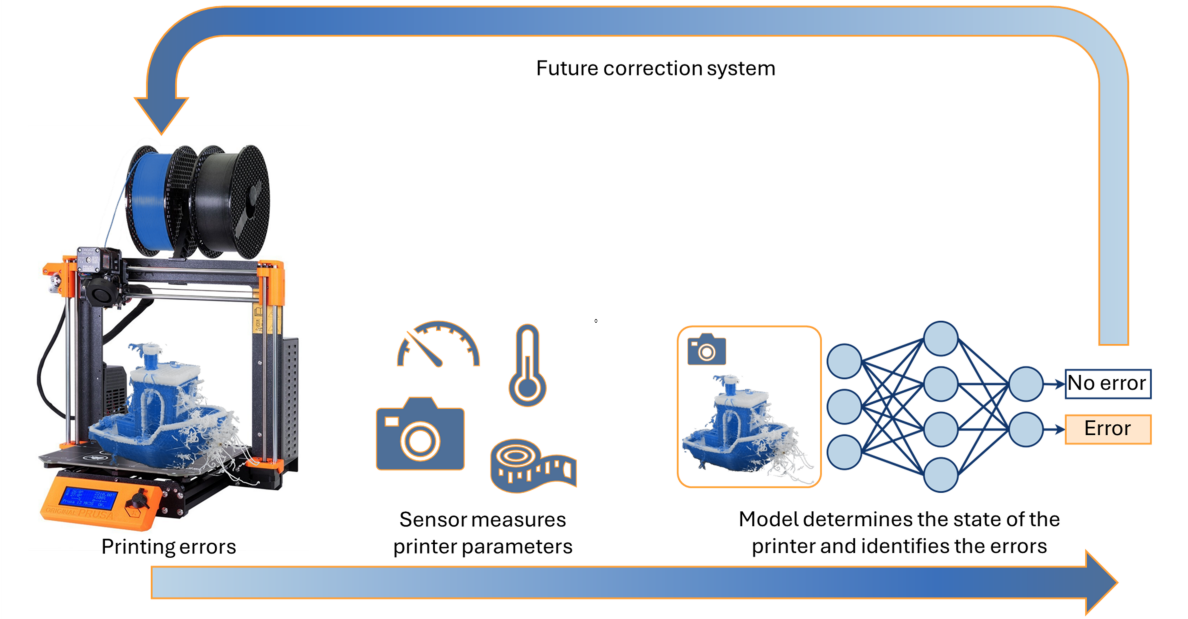


Figure 20.1: Fault detection system and correction system: the correction system aims to make the process closed-loop (adapted from [48, 96])

20.1. Future Steps to Achieving a Correction System

While the current system relies on classification models, future steps include incorporating object detection models. Object detection models not only classify the images but also provide information about the feature's location. Thus, object detection can predict the location of the error. Figure 20.2 shows what this object detection would look like. The capability to locate the features allows for more targeted and precise corrective actions, which is particularly useful in addressing localised errors. Furthermore, object detection allows for multiple errors to be detected within one image. It is worth noting that object detection is not strictly necessary for all cases. Parameter adjustments can still be made based solely on classification outputs, especially for global errors like over extrusion or under extrusion. Nonetheless, the inclusion of object detection adds robustness to the system. Its drawbacks, such as increased computational requirements and potentially more complex training processes, should also be considered when deciding on its implementation.

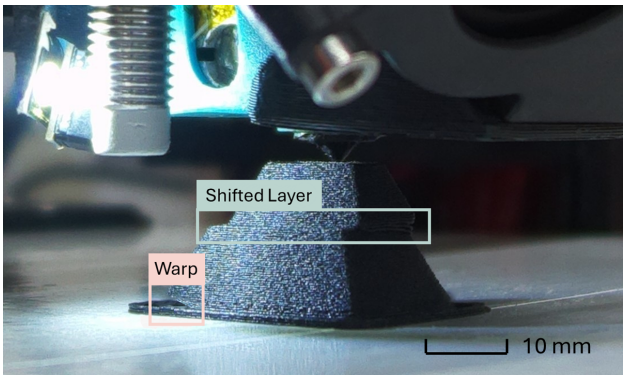


Figure 20.2: Example of how object detection could look like, where multiple errors and their location can be identified

Another part worth exploration is the fault severity estimation. This functionality involves quantifying the magnitude of a detected error, enabling more precise adjustments to printing parameters. For instance, if over extrusion is detected, the system should be able to estimate its degree. That would allow the system to reduce the extrusion rate appropriately. For minor issues, a small adjustment (e.g., a 10% reduction) might suffice, while more severe cases could require larger corrections.

The final and most significant step would then be the development of a correction mechanism. This mechanism should dynamically adjust printing parameters in response to detected errors, effectively closing the loop in the 3D printing process. The correction mechanism should ideally depend on the type of error, its localisation and severity.

20.2. Correction per Error Type

To mitigate or limit detectable errors during 3D printing, specific printing parameters can be adjusted based on the error type. The following subsections detail the appropriate corrective actions for each error category.

20.2.1. Extrusion Errors: Over Extrusion and Under Extrusion

Extrusion errors are among the most common issues in 3D printing and can significantly affect print quality. As demonstrated by Jin et al. [34], extrusion errors can be addressed by dynamically adjusting the following parameters: the print speed, flow rate and nozzle height. Slowing down the print speed can improve material flow consistency, particularly in under extrusion scenarios. Reducing the flow rate for over extrusion or increasing it for under extrusion can help balance material deposition. Modifying the nozzle height ensures proper layer adhesion and reduces material buildup in over extrusion cases. Implementing these adjustments in real time would mitigate extrusion-related issues and improve print consistency.

20.2.2. Finish Issues: Pillowing and Foreign Bodies

Pillowing occurs when gaps or uneven surfaces form on the top layer of a print. Since this issue only impacts the final appearance and strength of the top layer, it is critical to verify whether the error occurs on the topmost layer. If it does, the system could adapt the G-code (the print instructions) to add an additional layer, thereby enhancing the print's finish.

Foreign bodies on the print bed can cause surface defects or disrupt subsequent layers. Furthermore, they can result in aesthetic imperfections and compromise the print's overall quality. When detected, the system should halt printing and notify the operator to clean the print bed before resuming the print process. This ensures the print's surface integrity and reduces the risk of further issues.

20.2.3. Support and Structural Issues: Weak Infill and Poor Bridging

Weak infill compromises the structural properties of the printed part. When detected, the system should alert the user about the potential weakness. To address this issue, the G-code could be modified to increase the infill density or adjust printing parameters in real-time such as increasing the extrusion rate and and lowering print speed.

Poor bridging occurs when unsupported spans break during printing. This issue can be mitigated by increasing the fan speed to its maximum, which enhances cooling and strengthens the bridge. Additionally, the system should notify the operator to include support structures in the design to prevent bridging failures in future prints.

20.2.4. Temperature Issues: Stringing and Overhang Sagging

Stringing refers to thin strands of material left behind during travel movements. To minimise stringing, corrective actions can be taken. Increasing the retraction length ensures that more material is pulled back into the nozzle during travel, reducing oozing. In addition, increasing the travel speed decreases the time available for the material to ooze out. Moreover, the system should display a recommendation to placing the filament in a vacuum oven to eliminate moisture, which can worsen stringing.

Overhang sagging occurs when layers droop due to insufficient support or excessive heat. To address this issue when it is detected, the nozzle temperature should be reduced in real-time, to limit further drooping. Additionally, decreasing the print speed allows for improved layer cooling and adhesion, further preventing sagging. Furthermore the operator should receive a warning to add supports which will stabilise the overhang.

20.2.5. Layer Alignment Issues: Shifted Layers and Spaghetti Failure

Layer alignment issues, such as shifted layers and spaghetti failure, require immediate attention. When a shifted layer occurs, the printer should pause to allow the operator to assess the severity of the misalignment. The operator should be advised to inspect the printer for obstructions, perform maintenance if necessary and restart the print if the issue cannot be resolved.

Detection of spaghetti failure should result in an immediate stop to the printer. The system should then inform the operator of potential causes, such as failing supports or bed separation and provide recommendations to prevent similar failures in future prints.

20.2.6. Layer Adhesion Issues: Warping and Delamination

Warping, a print error where the edges of a print lift from the bed, can be mitigated by increasing the bed temperature, reducing the fan speed and lowering the lateral movement speed in real-time. Brion

et al. [35] created a fault detection and correction system for warping. The authors demonstrated that adjusting these parameters significantly limits warping, achieving notable improvements in print quality.

Delamination, which occurs when printed layers separate due to inadequate interlayer adhesion, could possibly be limited by lowering the fan speed real-time. Delamination frequently arises in tall designs or prints with sharp corners. Therefore, the system should prompt a message to the user advising to modify the design. Moreover, the fan speed should be lowered and print speeds should be reduced in real-time.

21| 45° Belt Printing

Belt printers introduce a unique approach to additive manufacturing by printing at a 45° angle. Unlike conventional 3D printers, belt printers replace the print bed with a conveyor belt. This belt, in perpetual motion, provides a continuously moving surface upon which the nozzle deposits material [97]. As a result, belt printers enable the production of ultra-long parts in one direction. This type over printer overcomes the size limitations typically imposed by the printer's frame. Figure 21.1 showcases such a 45° belt printer model [98].



Figure 21.1: Example of a 45° belt printer: Creality CR-30 Printmill (from [98])

The 45° belt printing technique offers several compelling advantages. By replacing the stationary print bed with a continuously moving conveyor belt, belt printers enable the production of parts with dimensions far beyond those of the printer itself. The continuous motion of the belt facilitates uninterrupted printing, as completed parts are automatically removed, paving the way for seamless production cycles and integration into automated manufacturing pipelines [97, 99, 100]. Additionally, the angled deposition of material reduces the challenges associated with overhangs, a common issue in conventional 90° printing.

However, the belt technology is not without limitations. The angled printing approach can compromise print quality, with challenges in achieving precise dimensional accuracy and fine surface finishes. Adhesion can also be problematic, as the belt material is often less optimal compared to conventional heated beds, restricting compatibility with certain materials. Belt printers are generally suitable for a narrower range of materials and are more expensive on average than traditional printers. This niche technology also suffers from limited community and developer support, as well as a lack of standardised solutions, and is considered to still be in the developmental stage [97, 100].

21.1. Relevance for 45° Printing

Despite the promising potential of 45° belt printing, research on this technique remains limited, particularly in the context of fault detection systems. A significant gap exists in the literature, as up to now, the literature is devoid of data that specifically address error detection in belt printing. This highlights an opportunity for further exploration and innovation in this area.

It is important to note that 45° belt printing shifts overhang-related challenges. This is illustrated in Figure 21.2, which compares overhang scenarios in 90° and 45° printing [99]. When building up layers under a 45° angle, the overhang also rotates 45° relative to the bottom layer. However, many print errors observed in conventional 90° printing still apply. These include extrusion errors, finish issues, support and structural errors, temperature-related errors, layer alignment problems and layer adhesion issues. As such, the implementation of an error detection system remains of utmost relevance for ensuring high-quality prints.

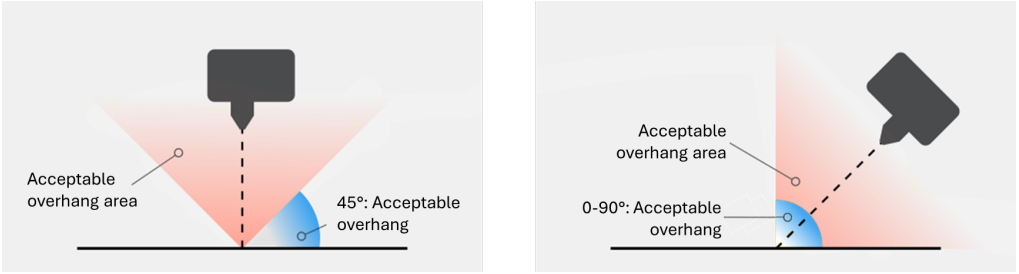


Figure 21.2: Overhang principle in 90° and 45° printing: which overhangs are acceptable differs from the printing angle [99]

21.2. Adjusting System to 45° Printers

To adapt the existing fault detection system for 45° belt printing, modifications are required in terms of camera placement and model calibration.

The side cameras can be positioned similarly to the original design, mounted on the frame of the print bed. These could be placed at the back of the printer, under a side angle. This allows for inspection of the side of the print and the new deposited layers. For the nozzle camera, a new custom bracket must be designed to accommodate the 45° angled nozzle. Proper visibility of both nozzle and deposited material is crucial to ensure accurate error detection. The bracket design should prioritise stability and precision to avoid obstructing the camera's field of view.

The existing convolutional neural network (CNN) models used for side and nozzle cameras must be tested for accuracy in detecting errors on a belt printer. If the models perform adequately, the system can be implemented directly. However, if performance issues arise, additional data collection and retraining may be necessary. Transfer learning offers an efficient approach to adapt the models for belt printing. By freezing the parameters of the initial layers and updating only the final layers with new data, the models can be fine-tuned to accommodate the unique characteristics of 45° belt printing. This method reduces training time while leveraging the knowledge gained from the original system.

The incorporation of fault detection systems in 45° belt printers has the potential to enhance reliability and efficiency in additive manufacturing. By addressing print errors specific to this technique, the system can contribute to the broader adoption of belt printing in both industrial and research settings. However, continued research and development will be necessary to fully realise this potential and bridge the existing literature gap.

22| Large-scale 3D printing

Large-scale 3D printing involves the production of objects significantly larger than those created by conventional desktop printers, often reaching dimensions measured in meters. This technique often employs specialised hardware including robot arms, capable of handling the demands of increased size, weight and material usage. One example is the 3D printing of complete leisure boats by the company Tanaruz, which uses a unique large-scale 3D printing system. This system involves a robotic arm mounted on a rail system that enables precise movement along the length of the print. The rail allows the robot to travel while its arms provide additional degrees of motion. Figure 22.1 illustrates Tanaruz' printing robot [1].

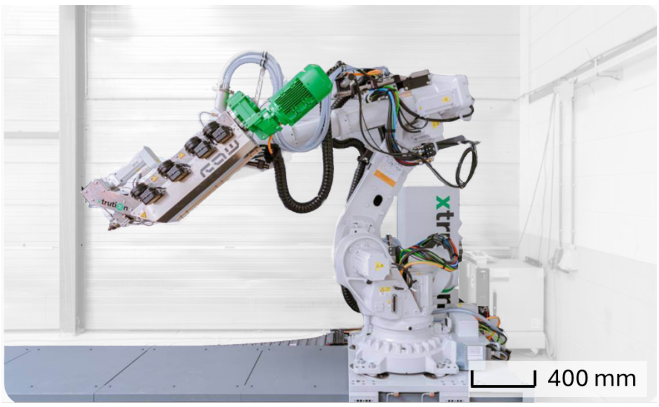


Figure 22.1: Large scale 3D printing robot (adapted from [1])

Large-scale 3D printing significantly differs from traditional 3D printing in several key aspects, as highlighted by CEAD [95]. Firstly, everything in large-scale printing is physically larger. For instance, Tanaruz prints boats up to 6.5 meters in length [1]. Consequently, the nozzle used in this process is also significantly larger, with a nozzle width of 10 [mm] compared to the much smaller diameters of 0.4 [mm] used in traditional desktop printers. This larger nozzle size allows for faster material deposition but results in thicker layers, often eliminating the need for internal infill structures.

Unlike smaller printers, large-scale systems typically avoid active cooling during printing. Cooling is considered undesirable as it can introduce thermal stresses and distortions in the large parts being fabricated. Furthermore, a continuous toolpath is desired to ensure seamless material deposition without interruptions. Halts in the process could lead to visible defects. This continuous approach, however, restricts the system to using a single material at a time. Support structures, which are a common requirement in smaller-scale 3D printing, are rarely used in large-scale printing. The 45° printing approach minimises overhang challenges, reducing the need for supports. Moreover, supports in large-scale projects is impractical due to the sheer size and weight of the parts, making their avoidance a priority.

Tanaruz employs a particularly novel printing methodology. The printer works airborne, meaning that as parts are being printed, their weight gradually causes them to shift downwards until they reach the ground for support. This innovative approach eliminates the need for additional supports but requires

precise control over the printing parameters to ensure stability. There is currently no literature on airborne FDM printing. This makes it a promising area for further exploration to optimise print quality, improve structural stability and expand its potential applications in large-scale manufacturing.

22.1. Relevance for Large-Scale Printing

Large-scale printing introduces unique challenges that differ from those encountered in traditional 3D printing. The errors that arise in large-scale projects are often linked to the cooling rate, material deposition and the mechanical properties of the printed parts. Unlike smaller-scale printing, where a visual error detection system can intervene during the process, the larger size and slower reaction times in large-scale printing systems make immediate correction more challenging. For example, visual systems may detect an error too late for effective correction due to the delayed cooling process. Nevertheless, introducing a fault detection system using computer vision could still be beneficiary. When severe errors are detected, the printed could be halted, limiting continuous monitoring by operators. This could considerably save valuable printing time, material and costs.

Some of the conventional errors observed in smaller printers such as issues with small features or intricate details are irrelevant for large-scale projects. Large-scale printing systems are more prone to errors such as over or under extrusion, poor surface quality or structural misalignments. These are worsened by the size and weight of the printed parts. Table 22.1 provides a summary of common errors and their relevance to large-scale printers [1]. The differences highlight the need for a tailored fault detection and correction system specifically designed for large-scale applications as compared to regular 3D printers. To achieve such tailored fault detection system, the same approach developed in this thesis could be applied to start with.

Error Group	Mistake	Big Printers?
Extrusion	Over extrusion	Yes
	Under extrusion	Yes
Details and Finish	Small feature not printing	No
	Foreign body in layer	Yes
	Pillowing	No
Support and Structures	Support falls apart	No
	Poor surface quality at support	Yes
	Weak infill	No
	Infill showing on exterior of print	Yes
	Poor bridging	No
Temperature and Cooling	Stringing	Yes
	Overheating and burnt blobs	Yes
	Overhang sagging	No
Layer Alignment	Missing layer	Yes
	Shifted layer and leaning prints	Yes
	Ripples and Echoes	Yes
	Spaghetti	Yes
Layer Adhesion	First layer issues	No
	Circles are not round	No
	Print not sticking to bed	Yes
	Warping	Yes
	Elephant foot	No
	Delamination & cracks in tall object	Yes

Table 22.1: Large-scale printer mistakes overview

22.2. Adjusting System to Large-Scale Printers

The fault detection system developed for small-scale printing can be scaled up for large-scale applications, but adjustments are required. The printer used by Tanaruz, an ABB6700 robot mounted on a rail system presents the opportunity to include such fault detection system. The robot prints exclusively on one side of the track, suggesting a practical mounting system for cameras is feasible, as illustrated in Figure 22.2. Cameras can be positioned along the rails to capture the entire printing area. A nozzle camera can be mounted directly on the robot arm to monitor material deposition in real-time. Due to the larger size of the nozzle and the greater distance between the nozzle and the camera, there is no need for specialised equipment such as borescopes.

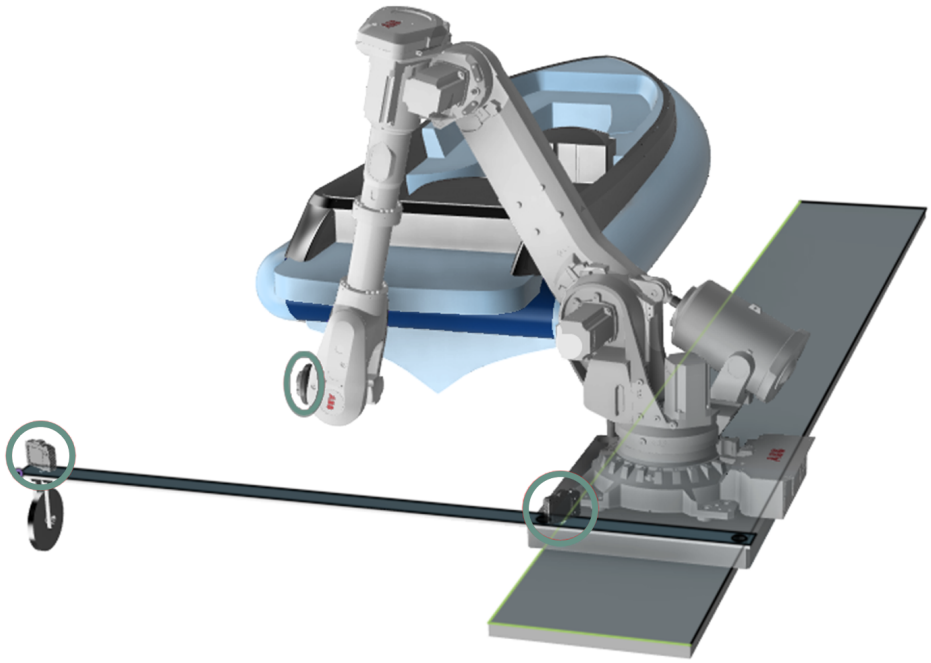


Figure 22.2: Large-scale printing setup of camera fault detection system, cameras are encircled in green (adapted from [47, 101, 102])

The fault detection models must be redesigned to address the unique characteristics of large-scale printing. These include the differences in error types, the larger scale and different material behaviour. This would ensure a robust system to handle the particular challenges posed by this novel approach to 3D printing.

23| Possible Impacts of the Fault Detection System

The implementation of an accurate fault detection system in 3D printing has the potential to deliver substantial benefits in terms of cost savings, resource efficiency and reduced environmental impact. This chapter aims to quantify the potential gains from implementing a fault detection system for both desktop-scale and large-scale 3D printers.

23.1. Desktop Printing

For desktop 3D printers the failure rate of prints is estimated to be around 20%, particularly when employed by inexperienced operators [28–31]. Many of these failures occur early in the printing process due to issues such as layer adhesion, extrusion inconsistencies or incorrect print bed levelling. An early fault detection could therefore significantly reduce material and energy waste. It can be estimated that up to 15% of the material, energy and printing time could be saved by an efficient early detection and automatic halting system.

Moreover, operators would no longer need to monitor printers continuously. An automated fault detection system that halts printing at the first sign of an error would allow operators to focus on other tasks. This could further lead to an improved productivity. When implemented on conveyor belt systems, as discussed in Chapter 21, the benefits are amplified because continuous printing becomes viable without the risk of material wastage due to undetected errors.

23.2. Large-Scale Printing

For large-scale printing operations, such as those conducted by Tanaruz [1], the savings from a fault detection system are even more significant due to the scale of materials and time involved. Tanaruz manufactures 3D-printed leisure boats, which weigh approximately 800 [kg] each. Based on data of 2024, the failure to success rate is 16:7. This means that, on average, three attempts are required to produce only one boat. On average, printing errors occur at 37.5% completion of the print. However they are up to now detected far too late. Errors are identified and halted around 50% of the print. Notably, this is a rough estimation, as detection times vary depending on factors such as whether printing occurs during daytime or nighttime. Although this is a rough estimation, it indicated undoubtedly that there is a considerable potential to optimise the production process by implementing an fault detection and halting system.

For example, if errors occurs at 37.5% of the print and is detected right away instead of at 50% completion, up to $2 * (50\% - 37.5\%) * 800 = 200$ [kg] of material per successful boat could be conserved. This represents a reduction in material waste of 25% for each boat. Additionally, the energy consumption for printing a successful boat is estimated to be 1050 [kWh], meaning that early fault detection could save $2 * 12.5\% * 1050 = 262.5$ [kWh] per completed boat.

The financial benefits primarily stem from the energy and material saved. These savings can be calculated using Equation 23.1, where $C_{kg} = 3.95€/kg$ represents the material cost per kilogram [44] and $C_{kWh} = 0.508€/kWh$ denotes the cost per kilowatt-hour [103]. This results in saving €923.35 per boat. Considering, Tanaruz printed 7 successful boats in 2024, this system would have resulted in annual saving of $€6463.45$.

$$\begin{aligned} \text{Saving per Boat} &= C_{kg} * \text{kg saved per boat} + C_{kWh} * \text{kWh saved per boat} \\ &= €3.95 * 200 + 0.508 * 262.5 \\ &= €790 + 133.35 \\ &= €923.35 \end{aligned} \tag{23.1}$$

In terms of time, printing one boat takes approximately 85 hours (3.5 days). Detecting errors early and stopping the printer could save up 21 hours of printing time for each successful boat. This could results in an increased production of 27.12%.

24| Significance of This Work

This research presents a novel approach to fault detection in FDM 3D printing by integrating multiple cameras and leveraging convolutional neural networks for automated error classification. The system introduces several key innovations, providing a foundation for further advancements in in-situ fault detection and potential correction mechanisms. This chapter highlights the main contributions and significance of this work.

24.1. Key Contributions

- **Comprehensive fault detection:** The system is capable of detecting 12 distinct 3D printing error types, offering a broad coverage of common failure modes.
- **Multi-camera integration:** A structured detection framework combines inputs from a nozzle camera and two bed cameras, enabling error detection from multiple viewpoints.
- **Neural network-based classification:** Two convolutional neural network (CNN) models are utilised to classify errors based purely on image data, demonstrating high classification accuracy.
- **Logic-based decision system:** A structured logic framework integrates data from all cameras to determine the most probable error type, improving reliability in ambiguous cases.
- **Standalone image-based approach:** This system relies solely on visual data, making it adaptable and independent.
- **Handling diverse error types:** The system successfully integrates visually distinct errors, such as over extrusion and delamination, into a unified detection framework, overcoming the limitations of previous research that focused on similar error groups.
- **Cross-camera data analysis:** The detection logic accounts for how an error appears across different camera perspectives, allowing for more robust identification, even when an error is only visible from one viewpoint.
- **Automated image preprocessing:** The system includes automated image cropping to match model input dimensions, ensuring consistency in classification.

24.2. Insights Gained

- **Interdependencies between errors:** Through visual analysis, relationships between errors were identified (e.g., overheating as a root cause for other errors, stringing could prevent poor bridges, over-extrusion leads to warping), contributing to a better understanding of failure mechanisms.
- **Camera suitability for fault detection:** The study provided insights into which camera types are most effective for monitoring different aspects of the print process.
- **Limitations of visual detection:** Certain errors, such as elephant foot, lack sufficient visible features to be reliably detected, while others require a reference model for comparison.

24.3. Broader Impact and Future Directions

The findings of this research provide a foundation for more advanced fault detection and potential correction mechanisms in 3D printing. By demonstrating that a high-accuracy model can be trained based on image-data, this work opens avenues for real-time monitoring systems that do not require extensive sensor integration. Furthermore, the insights gained about error interdependencies and camera effectiveness can guide future research in optimising detection frameworks. Expanding the dataset to include more materials and geometries will further enhance the system's applicability to diverse printing scenarios.

PART VIII

Conclusion

25| Conclusion

This thesis explores an innovative approach to fault detection in Fused Deposition Modelling (FDM). This a type of 3D printing that builds objects layer-by-layer using thermoplastic materials. While 3D printing offers tremendous flexibility, it is prone to errors that increase the cost and duration of the production process. Therefore, there is an unmet need of an effective fault detection system. The detection systems currently available are sub-optimal as they typically focus on specific error types or require the system to pause during error detection. The proposed system in this thesis integrates a multi-camera setup and convolutional neural networks (CNNs) models. The hypothesis was that this system enhances detection accuracy and cover more printing errors by capturing the printing process from multiple angles in real-time. To achieve this, the proposed system integrates data of three cameras: two side cameras focusing on the printer bed and one nozzle camera. Employing multiple cameras simultaneously requires an integration process. This involves combining results from different cameras, capable of detecting different error types. The innovative system represent a significant step towards creating smarter, more reliable FDM printing systems and lay the foundation for future research into integrating corrective mechanisms.

Two CNN models are developed, one for the nozzle camera and one for the side cameras. These models are trained to detect 12 common printing errors: over extrusion, under extrusion, foreign objects, pillowing, weak infill, poor bridging, stringing, spaghetti failure, overhang sagging, shifted layers, warping and delamination. The dataset used for training is diversified with various shapes, infill patterns and data augmentation techniques to ensure robustness.

The nozzle camera model achieves a validation accuracy of 97.68% and a validation loss of 0.075. This performance is consistent across training and test datasets. It achieves an F1 score of 0.9766, confirming its high-quality predictions. An external validation set is used to evaluate its generalisability. The model shows an accuracy of 93.18% for the external set, with only a slight reduction in performance. However, some classes, such as over extrusion and weak infill, showed lower recall and accuracy, likely due to dataset imbalances or small external validation subsets (e.g., 19 images for weak infill). Despite these challenges, the nozzle camera model demonstrates robust performance and generalisability, although minor improvements could be made to address class-specific weaknesses.

The side camera model achieves a validation accuracy of 97.61% and a validation loss of 0.1196, indicating high performance during training. When tested with an external dataset, the accuracy dropped to 89.3%, primarily due to reduced performance for certain classes like stringing, poor bridging and shifted layers. The main reason is that some images are misclassified as warping. Upon closer inspection, warping was often co-occurring with other errors. Additionally, overhang sag shows lower precision, likely because the external validation set included large overhang geometries. While the side camera model performs well overall, its generalisability could most probably be improved by expanding the dataset to better cover edge cases.

The two models are integrated into a unified fault detection system, a truly innovative step forward in fault detection for 3D printers. A logic framework is implemented to ensure that errors are consistently detected before being reported as faults. This logic also accounts for error types detectable by only one camera using prioritisation.

Although the system is designed for small-scale desktop printers, the methodology could be adapted for other printing techniques, such as 45° belt printing and even for large-scale industrial applications, possibly resulting in considerable savings in resources.

In summary, this work shows the feasibility of developing a robust multi-camera fault detection system to improve 3D printing. It paves the way for further research and implementation into complex integrated error correction mechanisms.

26| Future Recommendations

This thesis lays the foundation for an integrated fault detection system in FDM printing. There are several avenues for future research and development to further improve and expand its capabilities.

The first recommendation could be to create a more diversified dataset. Expanding the dataset to include a wider variety of colours, materials and shapes could significantly improve the robustness and generalisability of the detection models. By training on a broader range of data, the system could better handle real-world variations and edge cases encountered in diverse printing scenarios.

Secondly, it could be interesting to incorporate an error severity estimation. Future systems should not only classify errors but also assess their severity. Quantifying error severity could potentially allow for more precise interventions, such as determining whether a print should continue, pause or be aborted.

Thirdly, transitioning from classification to object detection could enable the system to localise errors within the print and detect multiple errors in a single image. This approach could provide more actionable insights by identifying specific regions of failure, enhancing the practicality of the detection system.

Fourthly, integrating a correction mechanism would elevate the system from fault detection to fault management. By modifying printing parameters in response to specific error types, the system could actively correct issues such as adjusting extrusion rates for over extrusion. This ensures a large impact with respect to printing time, resource waste and costs.

Fifthly, while this thesis focuses on desktop FDM printers, the methodologies developed here could be extended to other types of printers, such as large-scale industrial systems and 45° belt printers. Furthermore, exploring applications beyond FDM could broaden the impact of the system, making it relevant for other additive manufacturing processes.

Finally, this study observed that certain errors, such as stringing, may positively influence other error types, such as reducing poor bridging. Investigating these interactions could uncover additional relationships between errors and could lead to a further optimised printing strategy that exploit these benefits.

References

[1] RAW Idea. Tanaruz 3d-printed boats. https://tanaruz.boats/, 2024. Retrieved on March 19, 2024.

[2] Luca Rettenberger, Nils Beyer, Ingo Sieber, and Markus Reischl. Fault detection in 3d-printing with deep learning. In *2024 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–4, 2024. doi: 10.1109/ICCE59016.2024.10444198.

[3] Zeqing Jin, Zhizhou Zhang, Kahraman Demir, and Grace X. Gu. Machine learning for advanced additive manufacturing. *Matter*, 3(5): 1541–1556, 2020. ISSN 2590-2385. doi: https://doi.org/10.1016/j.matt.2020.08.023.

[4] Jahan Zeb Gul, Memoon Sajid, Muhammad Muqeet Rehman, Ghayas Uddin Siddiqui, Imran Shah, Kyung-Hwan Kim, Jae-Wook Lee, and Kyung Hyun Choi. 3d printing for soft robotics - a review. *Science and Technology of Advanced Materials*, 19(1):243–262, 2018. doi: 10.1080/14686996.2018.1431862. PMID: 29707065.

[5] Jigar Patadiya, Adwait Gawande, Ganapati Joshi, and Balasubramanian Kandasubramanian. Additive manufacturing of shape memory polymer composites for futuristic technology. *Industrial & Engineering Chemistry Research*, 60, 10 2021. doi: 10.1021/acs.iecr.1c03083.

[6] Aamir Ahmed, Sandeep Arya, Vinay Gupta, Hidemitsu Furukawa, and Ajit Khosla. 4d printing: Fundamentals, materials, applications and challenges. *Polymer*, 228:123926, 2021. ISSN 0032-3861. doi: https://doi.org/10.1016/j.polymer.2021.123926.

[7] Zia Ullah Arif, Muhammad Yasir Khalid, Ali Tariq, Mokarram Hossain, and Rehan Umer. 3d printing of stimuli-responsive hydrogel materials: Literature review and emerging applications. *Giant*, 17:100209, 2024. ISSN 2666-5425. doi: https://doi.org/10.1016/j.giant.2023.100209.

[8] Arthur Alexandre, Fabio A. Cruz Sanchez, Hakim Boudaoud, Mauricio Camargo, and Joshua M. Pearce. Mechanical properties of direct waste printing of polylactic acid with universal pellets extruder: Comparison to fused filament fabrication on open-source desktop three-dimensional printers. *3D Printing and Additive Manufacturing*, 7(5):237–247, 2020. doi: 10.1089/3dp.2019.0195.

[9] Muhammad Yasir Khalid, Zia Ullah Arif, Ali Tariq, Mokarram Hossain, Kamran Ahmed Khan, and Rehan Umer. 3d printing of magneto-active smart materials for advanced actuators and soft robotics applications. *European Polymer Journal*, 205:112718, 2024. ISSN 0014-3057. doi: https://doi.org/10.1016/j.eurpolymj.2023.112718.

[10] A. Mitchell, U. Lafont, M. Hołyńska, and C. Semprimoschnig. Additive manufacturing - a review of 4d printing and future applications. *Additive Manufacturing*, 24:606–626, 2018. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2018.10.038.

[11] Ryan T. Shafranek, S. Cem Millik, Patrick T. Smith, Chang-Uk Lee, Andrew J. Boydston, and Alshakim Nelson. Stimuli-responsive materials in additive manufacturing. *Progress in Polymer Science*, 93:36–67, 2019. ISSN 0079-6700. doi: https://doi.org/10.1016/j.progpolymsci.2019.03.002.

[12] Somayeh Vatanparast, Alberto Boschetto, Luana Bottini, and Paolo Gaudenzi. New trends in 4d printing: A critical review. *Applied Sciences*, 13(13), 2023. ISSN 2076-3417. doi: 10.3390/app13137744.

[13] Xueli Zhou, Luquan Ren, Zhengyi Song, Guiwei Li, Jifeng Zhang, Bingqian Li, Qian Wu, Wangxuan Li, Lei Ren, and Qingping Liu. Advances in 3d/4d printing of mechanical metamaterials: From manufacturing to applications. *Composites Part B: Engineering*, 254:110585, 2023. ISSN 1359-8368. doi: https://doi.org/10.1016/j.compositesb.2023.110585.

[14] Julien Gardan. Smart materials in additive manufacturing: state of the art and trends. *Virtual and Physical Prototyping*, 14(1):1–18, 2019. doi: 10.1080/17452759.2018.1518016.

[15] Annamaria Gisario, Michele Kazarian, Filomeno Martina, and Mehrshad Mehrpouya. Metal additive manufacturing in the commercial aviation industry: A review. *Journal of Manufacturing Systems*, 53:124–149, 2019. ISSN 0278-6125. doi: https://doi.org/10.1016/j.jmsy.2019.08.005.

[16] Ian Gibson, David Rosen, and Brent Stucker. Directed energy deposition processes. In *Additive Manufacturing Technologies: 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*, pages 245–268. Springer New York, New York, NY, 2015. ISBN 978-1-4939-2113-3. doi: 10.1007/978-1-4939-2113-3_10.

[17] Amal Megdich, Habibi Mohamed, and Luc Laperrière. A review on 4d printing: Material structures, stimuli and additive manufacturing techniques. *Materials Letters*, 337:133977, 02 2023. doi: 10.1016/j.matlet.2023.133977.

[18] Xometry, where big ideas are built. Polyjet vs. inkjet: Differences and comparison. https://www.xometry.com/resources/3d-printing/polyjet-vs-inkjet/, 2024. Retrieved on May 21, 2024.

[19] Wensong Ye, Haibing Li, Kang Yu, Chaoqi Xie, Peng Wang, Yating Zheng, Peng Zhang, Jingfang Xiu, Yi Yang, Feng Zhang, Yong He, and Qing Gao. 3d printing of gelatin methacrylate-based nerve guidance conduits with multiple channels. *Materials & Design*, 192:108757, 2020. ISSN 0264-1275. doi: https://doi.org/10.1016/j.matdes.2020.108757.

[20] Sciakie Inc. Electron beam additive manufacturing (ebam®). https://www.sciaky.com/additive-manufacturing/what-is-ded-3d-printing. Retrieved on May 17, 2024.

[21] Ajit Behera. Smart materials based additive manufacturing. In Shashanka Rajendrachari, editor, *Practical Implementations of Additive Manufacturing Technologies*, chapter 7, pages 153–175. Springer Nature, Singapore, 2024. doi: 10.1007/978-981-99-5949-5_7.

[22] Sophie Leanza, Shuai Wu, Xiaohao Sun, H. Jerry Qi, and Ruike Renee Zhao. Active materials for functional origami. *Advanced Materials*, 36(9), December 2023. ISSN 1521-4095. doi: 10.1002/adma.202302066.

[23] Prasansha Rastogi and Balasubramanian Kandasubramanian. Breakthrough in the printing tactics for stimuli-responsive materials: 4d printing. *Chemical Engineering Journal*, 366:264 – 304, 2019. doi: 10.1016/j.cej.2019.02.085. Cited by: 191.

[24] Donghwan Song, Adrian Matias Chung Baek, Jageon Koo, Moise Busogi, and Namhun Kim. Forecasting warping deformation using multivariate thermal time series and k-nearest neighbors in fused deposition modeling. *Applied Sciences*, 10(24), 2020. ISSN 2076-3417. doi: 10.3390/app10248951.

[25] Heime Jonkers, Alexander Dijkshoorn, Stefano Stramigioli, and Gijs Krijnen. Fff print defect characterization through in-situ electrical resistance monitoring. *Scientific Reports*, 14(1):11906, May 2024. ISSN 2045-2322. doi: 10.1038/s41598-024-59053-z.

[26] Douglas A. J. Brion and Sebastian W. Pattinson. Generalisable 3d printing error detection and correction via multi-head neural networks. *Nature Communications*, 13(1):4654, Aug 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-31985-y.

[27] Douglas A. J. Brion and Sebastian W. Pattinson. Quantitative and real□time control of 3d printing material flow through deep learning. *Advanced Intelligent Systems*, 4(11), September 2022. ISSN 2640-4567. doi: 10.1002/aisy.202200153. URL http://dx.doi.org/10.1002/aisy.202200153.

[28] Aliaksei L. Petsiuk and Joshua M. Pearce. Open source computer vision-based layer-wise 3d printing analysis. *Additive Manufacturing*, 36: 101473, 2020. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2020.101473.

[29] B.T. Wittbrodt, A.G. Glover, J. Laureto, G.C. Anzalone, D. Oppliger, J.L. Irwin, and J.M. Pearce. Life-cycle economic analysis of distributed manufacturing with open-source 3-d printers. *Mechatronics*, 23(6):713–726, 2013. ISSN 0957-4158. doi: https://doi.org/10.1016/j.mechatronics.2013.06.002.

[30] Aliaksei Petsiuk and Joshua M. Pearce. Towards smart monitored am: Open source in-situ layer-wise 3d printing image anomaly detection using histograms of oriented gradients and a physics-based rendering engine. *Additive Manufacturing*, 52, 2022. doi: 10.1016/j.addma.2022.102690.

[31] Siranee Nuchitprasitchai, Michael Roggemann, and Joshua M. Pearce. Factors effecting real-time optical monitoring of fused filament 3d printing. *Progress in Additive Manufacturing*, 2(3):133–149, Sep 2017. ISSN 2363-9520. doi: 10.1007/s40964-017-0027-x.

[32] Zeqing Jin, Zhizhou Zhang, and Grace X. Gu. Automated real□time detection and prediction of interlayer imperfections in additive manufacturing processes using artificial intelligence. *Advanced Intelligent Systems*, 2(1), December 2019. ISSN 2640-4567. doi: 10.1002/aisy.201900130.

[33] Zeqing Jin, Zhizhou Zhang, Joshua Ott, and Grace X. Gu. Precise localization and semantic segmentation detection of printing conditions in fused filament fabrication technologies using machine learning. *Additive Manufacturing*, 37:101696, 2021. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2020.101696.

[34] Zeqing Jin, Zhizhou Zhang, and Grace X. Gu. Autonomous in-situ correction of fused deposition modeling printers using computer vision and deep learning. *Manufacturing Letters*, 22:11–15, 2019. ISSN 2213-8463. doi: https://doi.org/10.1016/j.mfglet.2019.09.005.

[35] Douglas A.J. Brion, Matthew Shen, and Sebastian W. Pattinson. Automated recognition and correction of warp deformation in extrusion additive manufacturing. *Additive Manufacturing*, 56:102838, 2022. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2022.102838.

[36] Aditya Saluja, Jiarui Xie, and Kazem Fayazbakhsh. A closed-loop in-process warping detection system for fused filament fabrication using convolutional neural networks. *Journal of Manufacturing Processes*, 58:407–415, 2020. ISSN 1526-6125. doi: https://doi.org/10.1016/j.jmapro.2020.08.036.

[37] E. Ferraris, J. Zhang, and B. Van Hooreweder. Thermography based in-process monitoring of fused filament fabrication of polymeric parts. *CIRP Annals*, 68(1):213–216, 2019. ISSN 0007-8506. doi: https://doi.org/10.1016/j.cirp.2019.04.123.

[38] Jie Liu, Youmin Hu, Bo Wu, and Yan Wang. An improved fault diagnosis approach for fdm process with acoustic emission. *Journal of Manufacturing Processes*, 35:570–579, 2018. ISSN 1526-6125. doi: https://doi.org/10.1016/j.jmapro.2018.08.038.

[39] *A New Approach for Online Monitoring of Additive Manufacturing Based on Acoustic Emission*, volume Volume 3: Joint MSEC-NAMRC Symposia of *International Manufacturing Science and Engineering Conference*, 06 2016. doi: 10.1115/MSEC2016-8551.

[40] Haixi Wu, Zhonghua Yu, and Yan Wang. Real-time fdm machine condition monitoring and diagnosis based on acoustic emission and hidden semi-markov model. *The International Journal of Advanced Manufacturing Technology*, 90(5):2027–2036, May 2017. ISSN 1433-3015. doi: 10.1007/s00170-016-9548-6.

[41] Pascal Becker, Christian Roth, Arne Roennau, and Ruediger Dillmann. Acoustic anomaly detection in additive manufacturing with long short-term memory neural networks. In *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 921–926, 2020. doi: 10.1109/ICIEA49774.2020.9102002.

[42] Alexander Oleff, Benjamin Küster, Malte Stonis, and Ludger Overmeyer. Process monitoring for material extrusion additive manufacturing: a state-of-the-art review. *Progress in Additive Manufacturing*, 6(4):705–730, Dec 2021. ISSN 2363-9520. doi: 10.1007/s40964-021-00192-4.

[43] Prusa. Prusa mk3s. https://www.prusa3d.com/product/original-prusa-i3-mk3-s-to-mk4-upgrade-kit/, 2024. Retrieved on December 5, 2024.

[44] Transmare. Cirqlair. https://transmarecirqlair.com/, 2024. Retrieved on August 13, 2024.

[45] 3djake. 3d printer nozzle guide. https://www.3djake.com/info/guide/3d-printer-nozzle-guide, 2024. Retrieved on December 31, 2024.

[46] 3D Benchy. The jolly 3d printing torture-test. https://www.3dbenchy.com/, 2024. Retrieved on July 2, 2024.

[47] Paul Gherghisan. Grabcad community, simple photo camera. https://grabcad.com/library/simple-photo-camera-1, May 2024. Retrieved on December 25, 2024.

[48] 3DPrinterGear. Original prusa i3 mk3s+ assembled. https://www.3dprintergear.com.au/original-prusa-i3-mk3s-assembled, 2024. Retrieved on December 28, 2024.

[49] 3do. Enclosure nozzle camera v2 kit. https://3do.dk/3do-kamera/2333-3do-enclosure-nozzle-camera-v2-kit.html, 2024. Retrieved on August 9, 2024.

[50] Raspberry Pi. High quality camera. https://www.raspberrypi.com/products/raspberry-pi-high-quality-camera/, 2024. Retrieved on August 9, 2024.

[51] Raspberry Pi. High quality camera lense 6mm 3mp wide angle. https://thepihut.com/products/raspberry-pi-high-quality-camera-lens, 2024. Retrieved on December 7, 2024.

[52] Ivy Liam Gadgets. Raspberry pi hq camera 6mm 3mp lens (cs-mount). https://shop.ivyliam.com/product/6mm-3mp-lens-cs-mount/, 2024. Retrieved on December 8, 2024.

[53] Printables. Derek_184700, octopi camera bed mount for i3mk3s+. https://www.printables.com/model/125877-octopi-camera-bed-mount-for-i3mk3s, 2024. Retrieved on December 8, 2024.

[54] Raspberry Pi. Raspberry pi 3 model b. https://www.raspberrypi.com/products/raspberry-pi-3-model-b/, 2024. Retrieved on December 8, 2024.

[55] OctoPrint. The snappy web interface for your 3d printer. https://octoprint.org/, 2024. Retrieved on August 13, 2024.

[56] Raspberry Pi. Raspberry pi 5. https://www.raspberrypi.com/products/raspberry-pi-5/, 2024. Retrieved on December 8, 2024.

[57] Electronica voor jou. Raspberry pi 5 4gb. https://elektronicavoorjou.nl/en/product/raspberry-pi-5-4gb, 2024. Retrieved on December 28, 2024.

[58] Kiwi-electronics. Raspberry pi 3 b+. https://www.kiwi-electronics.com/nl/raspberry-pi-3-model-bplus-3392, 2024. Retrieved on December 28, 2024.

[59] Sourced. 44 common 3d print problems - troubleshooting issues 2023. https://www.3dsourced.com/rigid-ink/ultimate-3d-printing-troubleshooting-guide/, 2023. Retrieved on August 1, 2024.

[60] All3DP. 3d printing troubleshooting: All problems and solutions. https://all3dp.com/1/common-3d-printing-problems-troubleshooting-3d-printer-issues/, 2024. Retrieved on August 1, 2024.

[61] MatterHackers. 3d printer troubleshooting guide. https://www.matterhackers.com/articles/3d-printer-troubleshooting-guide#Issue8, 2024. Retrieved on August 1, 2024.

[62] Prusa Research by Josef Prusa. Prusaslicer 2.8.1. https://www.prusa3d.com/page/prusaslicer_424/, 2024. Retrieved on December 9, 2024.

[63] Thibaut Lucas. Picsellia: How to ensure image dataset quality in image classification. https://www.picsellia.com/post/image-data-quality-for-image-classification, January 2022. Retrieved on December 21, 2024.

[64] Indranil Balki, Afsaneh Amirabadi, Jacob Levman, Anne Martel, Žiga Emeršič, Blaž Meden, Angel Garcia-Pedrero, Saúl Calderón Ramírez, Dehan Kong, Alan Moody, and Pascal Tyrrell. Sample-size determination methodologies for machine learning in medical imaging research: A systematic review. *Canadian Association of Radiologists Journal*, 70, 09 2019. doi: 10.1016/j.carj.2019.06.002.

[65] Alexander S. Gilles. Techtarget: data splitting. https://www.techtarget.com/searchenterpriseai/definition/data-splitting, June 2024. Retrieved on December 21, 2024.

[66] Aradhna Saini, Gaurav Dhuriya, Ayush Jain, and Attiuttama Mishra. *Machine Learning Algorithms and Applications*, pages 1–31. 11 2024. ISBN 9781003504900. doi: 10.1201/9781003504900-1.

[67] Aryan Rose. How do artificial neural networks work. *Journal of Advances in Science and Technology*, 20:172 – 177, March 2023. ISSN 2230-9659. doi: https://doi.org/10.29070/ttrkmm98.

[68] Himanshi Singh. Analytics vidhya: Deep learning 101: Beginners guide to neural network. https://www.analyticsvidhya.com/blog/2021/03/basics-of-neural-network/, July 2023. Retrieved on December 31, 2024.

[69] Inc Cloud Software Group. Spotfire, neural networks explained: A look at functions applications. https://www.spotfire.com/glossary/what-is-a-neural-network, 2024. Retrieved on December 13, 2024.

[70] Poonam Rijwani, Sonal Jain, and Research Scholar. A genetic neural network based application for improving software development effort estimation. 07 2018.

[71] Andrea Apicella, Francesco Donnarumma, Francesco IsgrÃ², and Roberto Prevete. A survey on modern trainable activation functions. *Neural Networks*, 138:14–32, 2021. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2021.01.026. URL https://www.sciencedirect.com/science/article/pii/S0893608021000344.

[72] Rukshan Pramoditha. Medium, the concept of artificial neurons (perceptrons) in neural networks. https://towardsdatascience.com/the-concept-of-artificial-neurons-perceptrons-in-neural-networks-fab22249cbfc, December 2021. Retrieved on December 13, 2024.

[73] Amir Ali. Medium: Convolutional neural network(cnn) with practical implementation. https://medium.com/machine-learning-researcher/convlutional-neural-network-cnn-2fc4faa7bb63, May 2019. Retrieved on December 31, 2024.

[74] MRI Questions. Types of deep neural networks: What are the various types of deep networks and how are they used? https://mriquestions.com/deep-network-types.html#/, 2024. Retrieved on December 13, 2024.

[75] Yanqi Chen. The investigation of performance comparison for vgg, yolo, and dino in image classification. *Highlights in Science, Engineering and Technology*, 85:984–990, 03 2024. doi: 10.54097/9bgem219.

[76] Pinxue Lin. Comparative analysis of resnet50 with fully connected layers and yolov8 for object detection in cow stall number detection tasks, 05 2023.

[77] Vaibhav Patel, Mahendra Kanojia, and Vainavi Nair. Exploring the potential of resnet50 and yolov8 in improving breast cancer diagnosis: A deep learning perspective. *International Journal of Computer Information Systems and Industrial Management Applications*, 16(3):16, Jul. 2024.

[78] Shun ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4):185–196, 1993. ISSN 0925-2312. doi: https://doi.org/10.1016/0925-2312(93)90006-O. URL https://www.sciencedirect.com/science/article/pii/092523129390006O.

[79] Roseline Oluwaseun Ogundokun, Rytis Maskeliunas, Sanjay Misra, and Robertas Damaševičius. Improved cnn based on batch normalization and adam optimizer. In Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Ana Maria A. C. Rocha, and Chiara Garau, editors, *Computational Science and Its Applications – ICCSA 2022 Workshops*, pages 593–604, Cham, 2022. Springer International Publishing. ISBN 978-3-031-10548-7.

[80] Yixing Huang, Ahmed Gomaa, Daniel Höfler, Philipp Schubert, Udo Gaipl, Benjamin Frey, Rainer Fietkau, Christoph Bert, and Florian Putz. Principles of artificial intelligence in radiooncology. *Strahlentherapie und Onkologie*, Aug 2024. ISSN 1439-099X. doi: 10.1007/s00066-024-02272-0. URL https://doi.org/10.1007/s00066-024-02272-0.

[81] damian0815. Medium: Fine-tuning stable diffusion with validation. https://medium.com/@damian0815/fine-tuning-stable-diffusionwith-validation-3fe1395ab8c3, April 2023. Retrieved on December 15, 2024.

[82] Imad Dabbura. Coding neural network — dropout. https://towardsdatascience.com/coding-neural-network-dropout-3095632d25ce, May 2018. Retrieved on December 15, 2024.

[83] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[84] Aysenur Gilik, Selcuk Ogrenci, and Atilla Ã–zmen. Air quality prediction using cnn+lstm-based hybrid deep learning architecture. *Environmental Science and Pollution Research*, 29:1–19, 02 2022. doi: 10.1007/s11356-021-16227-w.

[85] Elliot Waite. Desmos: Softmax bar chart. https://www.desmos.com/calculator/drqqhtb037?lang=nl, 2020. Retrieved on December 28, 2024.

[86] Nyoman Bogi Aditya Karna, Made Adi Paramartha Putra, Syifa Maliah Rachmawati, Mideth Abisado, and Gabriel Avelino Sampedro. Toward accurate fused deposition modeling 3d printer fault detection using improved yolov8 with hyperparameter optimization. *IEEE Access*, 11: 74251 – 74262, 2023. doi: 10.1109/ACCESS.2023.3293056. Cited by: 18; All Open Access, Gold Open Access.

[87] Hyungjung Kim, Hyunsu Lee, Ji-Soo Kim, and Sung-Hoon Ahn. Image-based failure detection for material extrusion process using a convolutional neural network. *The International Journal of Advanced Manufacturing Technology*, 111(5):1291–1302, Nov 2020. ISSN 1433-3015. doi: 10.1007/s00170-020-06201-0. URL https://doi.org/10.1007/s00170-020-06201-0.

[88] Yanzhou Fu, Austin R.J. Downey, Lang Yuan, and Hung-Tien Huang. Real-time structural validation for material extrusion additive manufacturing. *Additive Manufacturing*, 65:103409, 2023. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2023.103409. URL https://www.sciencedirect.com/science/article/pii/S2214860423000222.

[89] Yuanbin Wang, Jiakang Huang, Yuan Wang, Sihang Feng, Tao Peng, Huayong Yang, and Jun Zou. A cnn-based adaptive surface monitoring system for fused deposition modeling. *IEEE/ASME Transactions on Mechatronics*, 25(5):2287–2296, 2020. doi: 10.1109/TMECH.2020.2996223.

[90] Ikenna A. Okaro, Sarini Jayasinghe, Chris Sutcliffe, Kate Black, Paolo Paoletti, and Peter L. Green. Automatic fault detection for laser powder-bed fusion using semi-supervised machine learning. *Additive Manufacturing*, 27:42–53, 2019. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2019.01.006.

[91] Mangolika Bhattacharya, Mihai Penica, Eoin O'Connell, and Martin Hayes. Ai-driven real-time failure detection in additive manufacturing. *Procedia Computer Science*, 232:3229–3238, 2024. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2024.02.138. URL https://www.sciencedirect.com/science/article/pii/S1877050924003168. 5th International Conference on Industry 4.0 and Smart Manufacturing (ISM 2023).

[92] Zhicheng Zhang, Ismail Fidan, and Michael Allen. Detection of material extrusion in-process failures via deep learning. *Inventions*, 5(3), 2020. ISSN 2411-5134. doi: 10.3390/inventions5030025. URL https://www.mdpi.com/2411-5134/5/3/25.

[93] Yaser Banadaki, Nariman Razaviarab, Hadi Fekrmandi, and Safura Sharifi. Toward enabling a reliable quality monitoring system for additive manufacturing process using deep convolutional neural networks, 2020. URL https://arxiv.org/abs/2003.08749.

[94] Hermann Baumgartl, Josef Tomas, Ricardo Buettner, and Markus Merkel. A deep learning-based model for defect detection in laser-powder bed fusion using in-situ thermographic monitoring. *Progress in Additive Manufacturing*, 09 2020. doi: 10.1007/s40964-019-00108-3.

[95] CEAD. Guidelines for large format 3d printing: from desktop prototyping to large format additive manufacturing. https://ceadgroup.com/guidelines-for-large-format-3d-printing/, April 2024. Retrieved on December 26, 2024.

[96] Beau Jackson. 3d printing industry: The 3d benchy 'torture test' that pushes 3d printers to the limit. https://3dprintingindustry.com/news/3d-benchy-torture-test-pushes-3d-printers-limit-103662/, 20197. Retrieved on January 30, 2025.

[97] Benedict O'Neill. Wevolver: Belt 3d printers explained: Technology, best machines, advantages. https://www.wevolver.com/article/belt-3d-printer, September 2022. Retrieved on December 26, 2024.

[98] Normax 3D. Creality cr-30 printmill belt printer. https://nm3d.no/produkt/creality-cr-30-printmill-belt-printer/, 2024. Retrieved on December 25, 2024.

[99] Adam Fasnacht. Powerbelt 3d: How to orient & slice models for a conveyor belt 3d printer. https://powerbelt3d.com/how-to-orient-models-for-a-conveyor-belt-3d-printer/, 2024. Retrieved on December 26, 2024.

[100] Jackson O'Connell. All3dp: Belt 3d printer: All you need to know. https://all3dp.com/2/3d-printer-belt-all-you-need-to-know/, May 2022. Retrieved on December 26, 2024.

[101] Manos. Grabcad community, rib. https://grabcad.com/library/rib--3, May 2012. Retrieved on December 25, 2024.

[102] Luca Delbarba. Grabcad community, abb anthropomorphic robots. https://grabcad.com/library/abb-anthropomorphic-robots-1, February 2022. Retrieved on December 25, 2024.

[103] Bluetti Power. Electricity price netherlands. https://www.bluettipower.eu/blogs/news/electricity-price-netherlands, 2023. Retrieved on December 28, 2024.

A| Confusion Matrices

A.1. Nozzle Camera Model

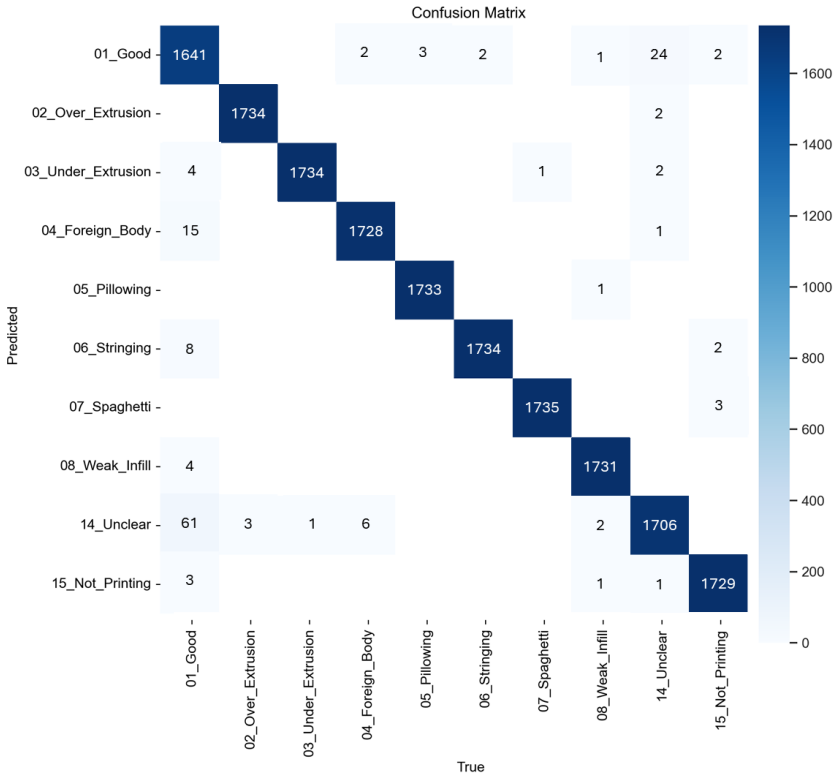


Figure A.1: Confusion Matrix Nozzle Camera, Training Data

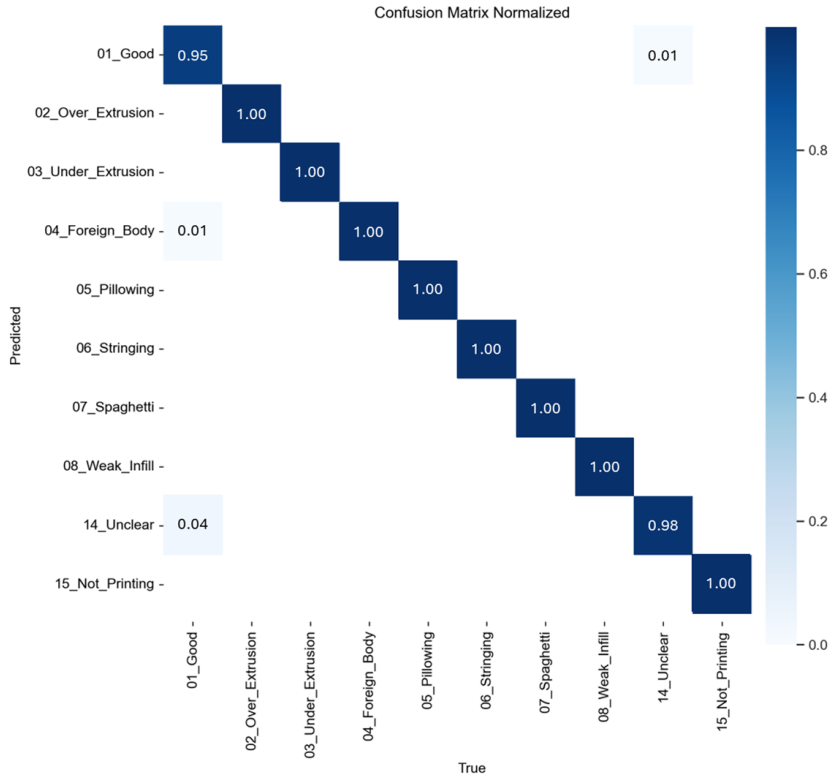


Figure A.2: Normalised Confusion Matrix Nozzle Camera, Training Data

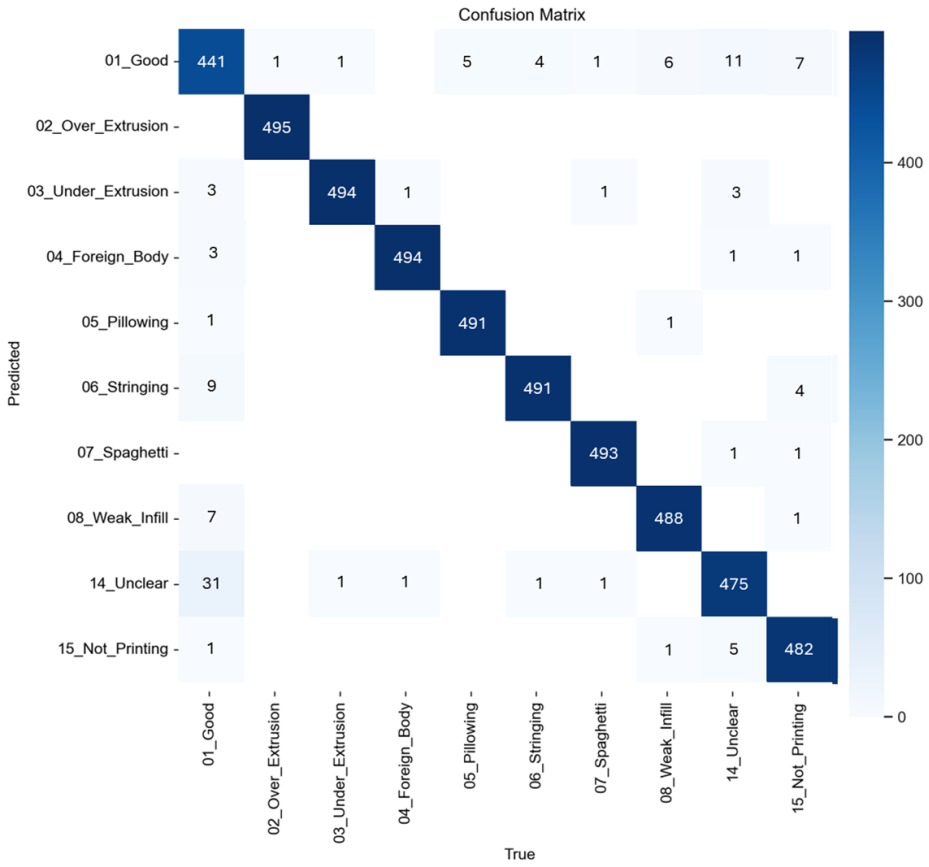


Figure A.3: Confusion Matrix Nozzle Camera, Validation Data

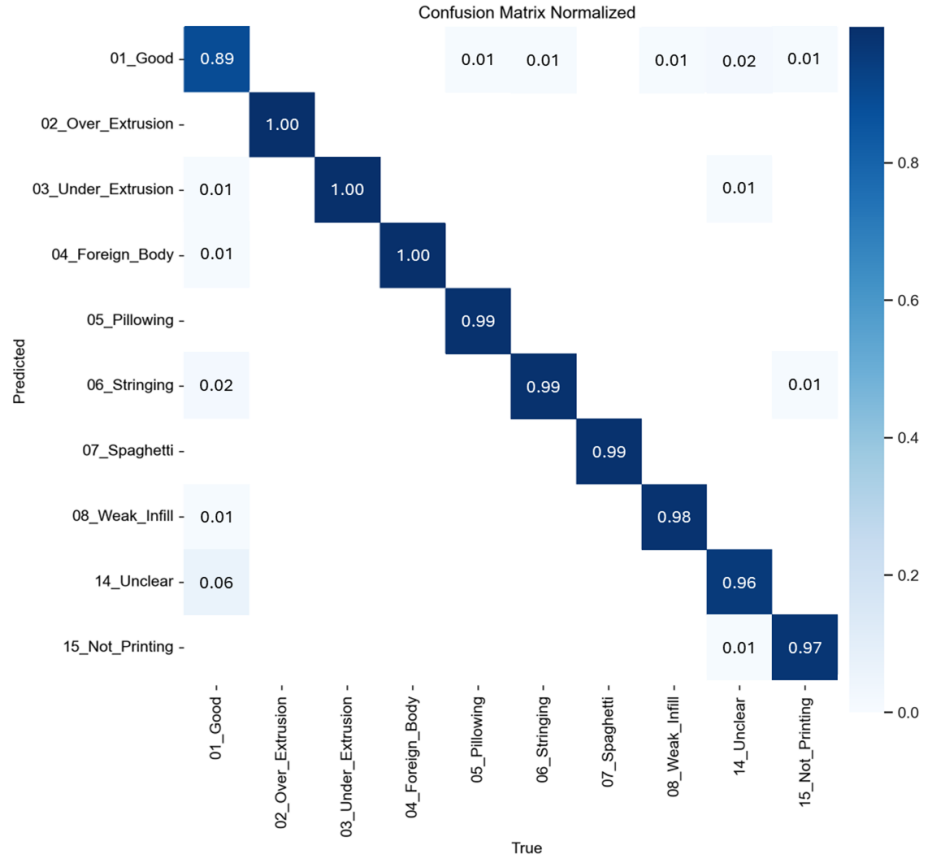


Figure A.4: Normalised Confusion Matrix Nozzle Camera, Validation Data

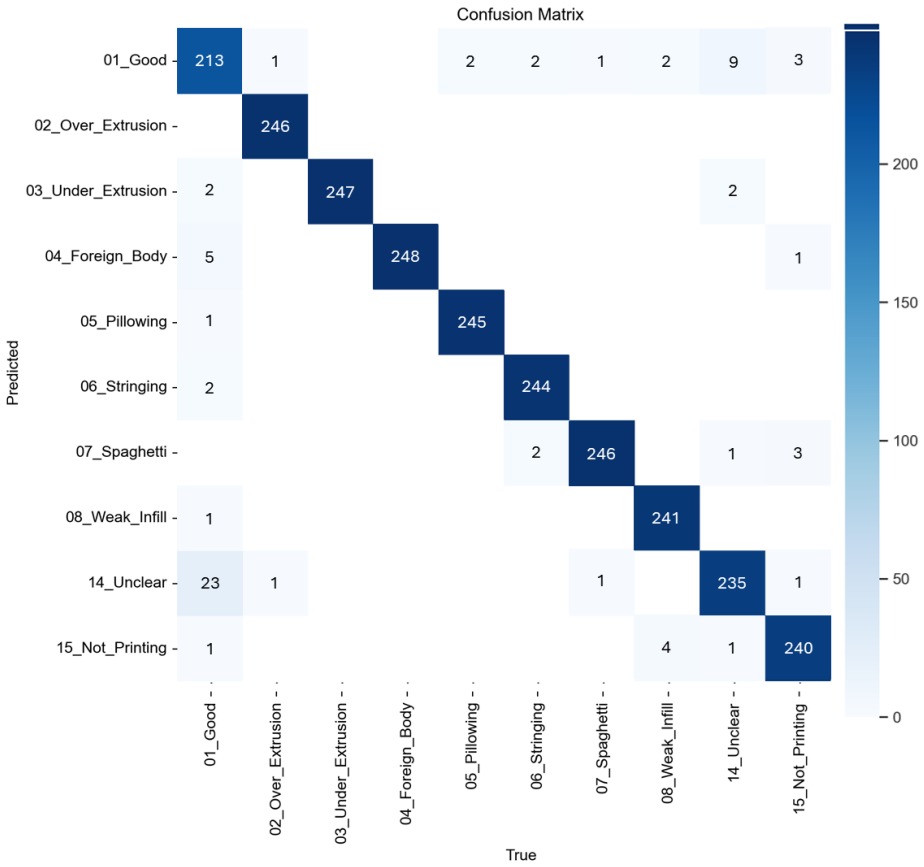


Figure A.5: Confusion Matrix Nozzle Camera, Test Data

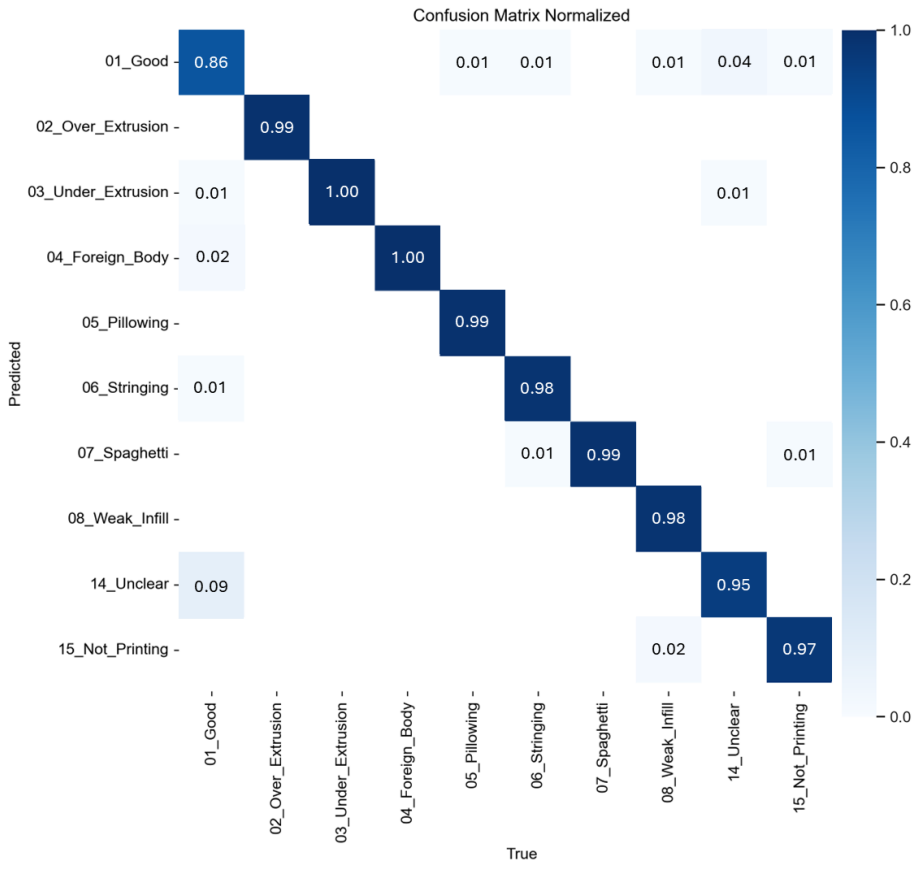


Figure A.6: Normalised Confusion Matrix Nozzle Camera, Test Data

A.2. Side Cameras Model

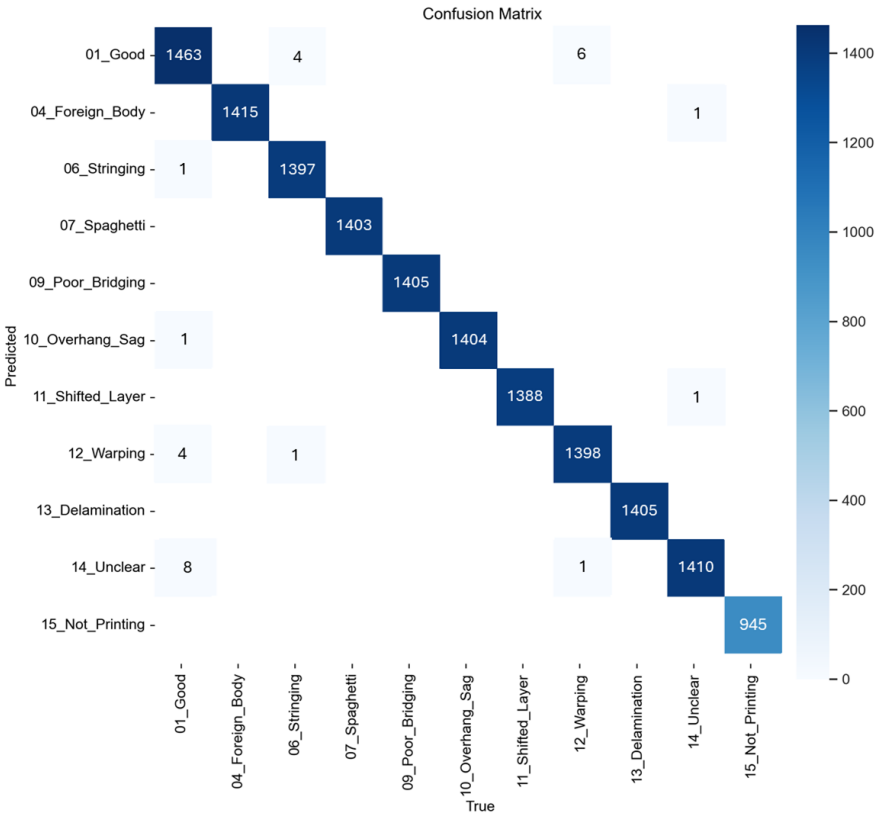


Figure A.7: Confusion Matrix Side Camera, Training Data

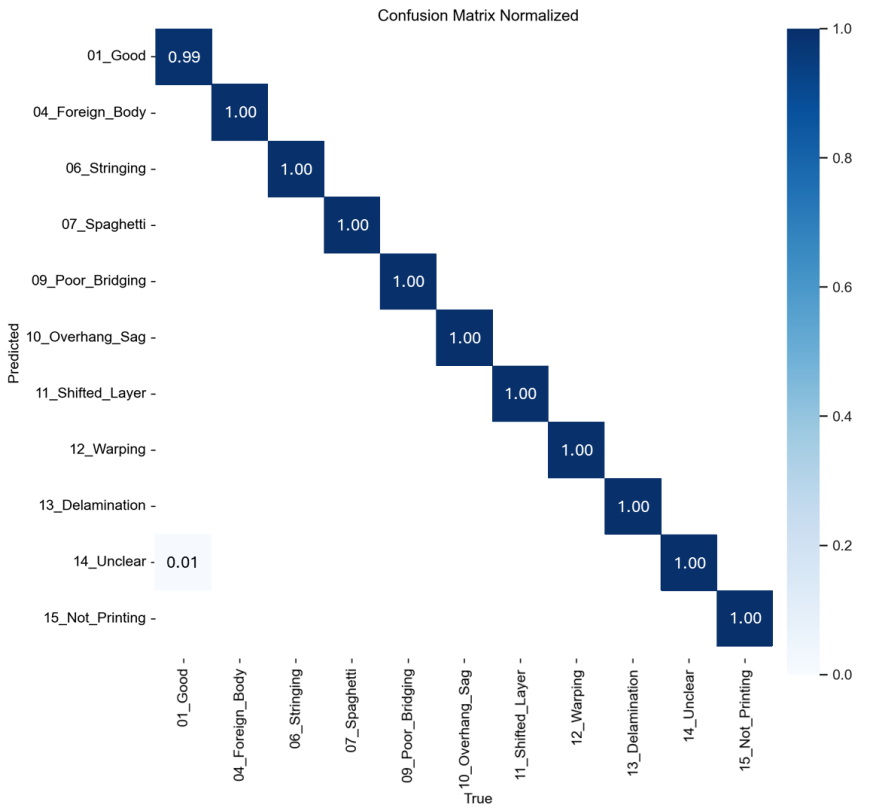


Figure A.8: Normalised Confusion Matrix Side Camera, Training Data

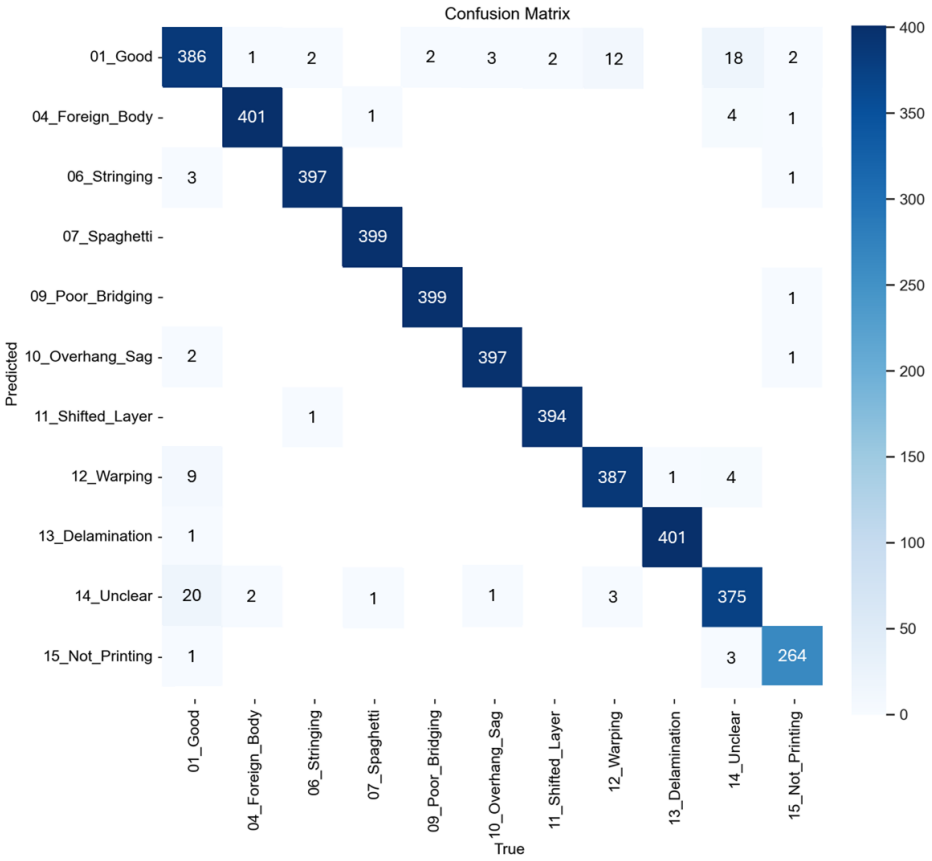


Figure A.9: Confusion Matrix Side Camera, Validation Data

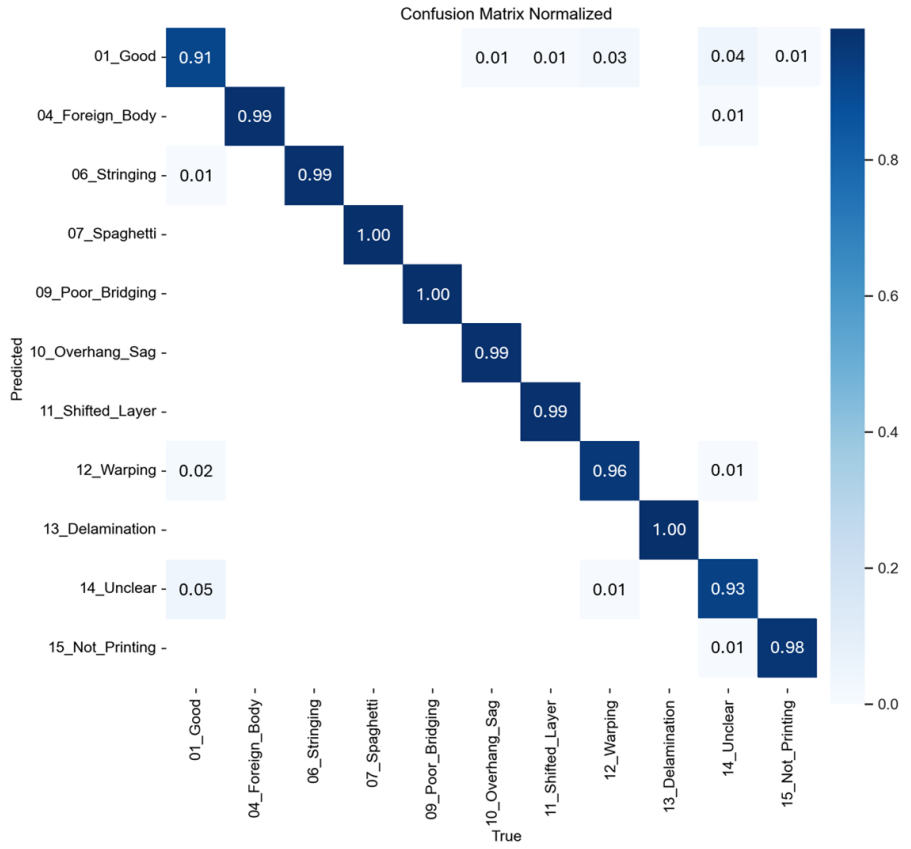


Figure A.10: Normalised Confusion Matrix Side Camera, Validation Data

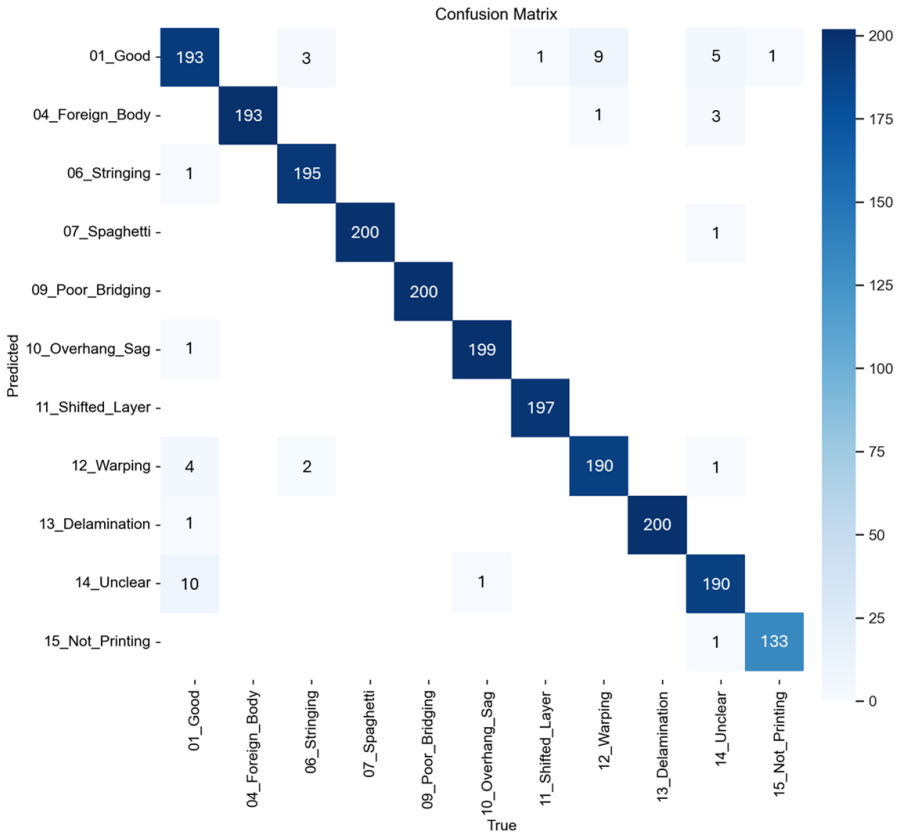


Figure A.11: Confusion Matrix Side Camera, Test Data

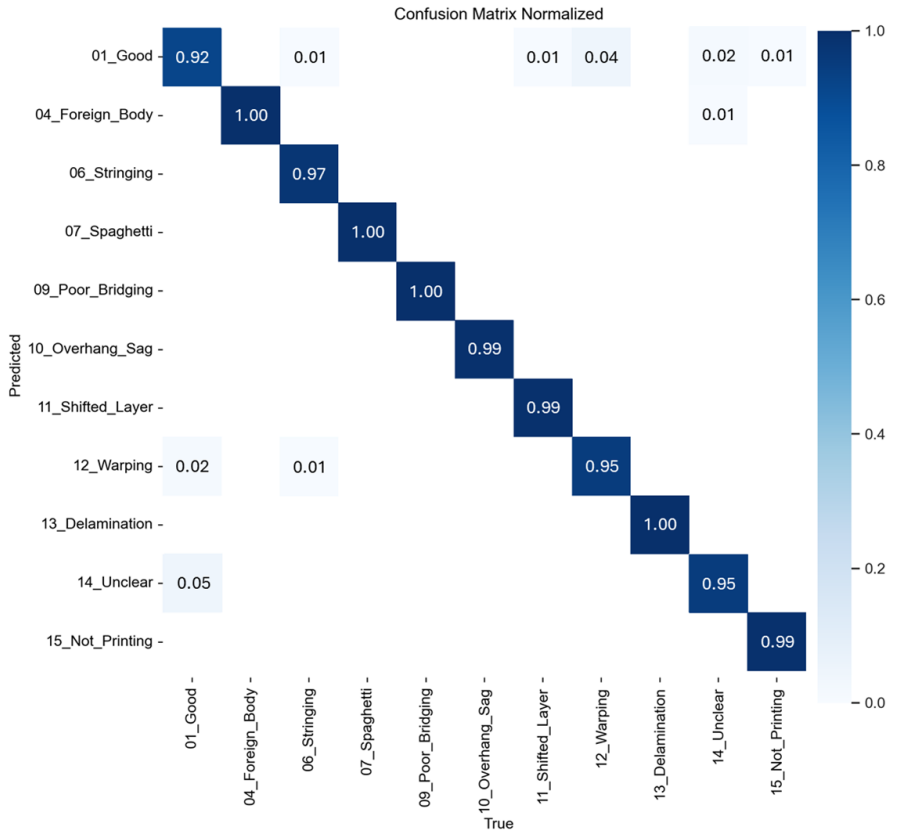


Figure A.12: Normalised Confusion Matrix Side Camera, Test Data

B| Precision, Recall and F1 Scores

B.1. Nozzle Camera Model

Class	Precision	Recall
Good	0.9797	0.9453
Over Extrusion	0.9988	0.9983
Under Extrusion	0.9960	0.9994
Foreign Body	0.9908	0.9954
Pillowing	0.9994	0.9983
Stringing	0.9943	0.9988
Spaghetti	0.9983	0.9994
Weak Infill	0.9977	0.9971
Unclear	0.9590	0.9827
Not Printing	0.9971	0.9960
Total	0.9911	0.9911
F1	**0.9911**	

Table B.1: Precision, Recall and F1 Score of Training Data of Nozzle Camera Model

Class	Precision	Recall
Good	0.9245	0.8891
Over Extrusion	1	0.9980
Under Extrusion	0.9841	0.9960
Foreign Body	0.9900	0.9960
Pillowing	0.9959	0.9899
Stringing	0.9742	0.9899
Spaghetti	0.9960	0.9940
Weak Infill	0.9839	0.9839
Unclear	0.9314	0.9577
Not Printing	0.9857	0.9718
Total	0.9766	0.9766
F1	**0.9766**	

Table B.2: Precision, Recall and F1 Score of Validation Data of Nozzle Camera Model

Class	Precision	Recall
Good	0.9142	0.8590
Over Extrusion	1	0.9919
Under Extrusion	0.9841	1
Foreign Body	0.9764	1
Pillowing	0.9959	0.9919
Stringing	0.9919	0.9839
Spaghetti	0.9762	0.9919
Weak Infill	0.9959	0.9757
Unclear	0.9004	0.9476
Not Printing	0.9756	0.9677
Total	0.9710	0.9710
F1	**0.9710**	

Table B.3: Precision, Recall and F1 Score of Test Data of Nozzle Camera Model

B.2. Side Cameras Model

Class	Precision	Recall
Good	0.9932	0.9905
Foreign Body	0.9993	1
Stringing	0.9993	0.9964
Spaghetti	1	1
Poor Bridging	1	1
Overhang Sag	0.9993	1
Shifted Layer	0.9993	1
Warping	0.9964	0.9950
Delamination	1	1
Unclear	0.9937	0.9986
Not Printing	1	1
Total	0.9982	0.9982
F1	**0.9982**	

Table B.4: Precision and Recall of Train Data of Side Camera Model

Class	Precision	Recall
Good	0.9019	0.9147
Foreign Body	0.9853	0.9926
Stringing	0.9900	0.9925
Spaghetti	1	0.9925
Poor Bridging	0.9975	0.9950
Overhang Sag	0.9925	0.9900
Shifted Layer	0.9975	0.9949
Warping	0.9651	0.9627
Delamination	0.9950	0.9975
Unclear	0.9328	0.9282
Not Printing	0.9851	0.9778
Total	0.9766	0.9762
F1	**0.9764**	

Table B.5: Precision and Recall of Validation Data of Side Camera Model

Class	Precision	Recall
Good	0.9104	0.9190
Foreign Body	0.9797	1
Stringing	0.9949	0.9750
Spaghetti	0.9950	1
Poor Bridging	1	1
Overhang Sag	0.9950	0.9950
Shifted Layer	1	0.9949
Warping	0.9645	0.9500
Delamination	0.9950	1
Unclear	0.9453	0.9453
Not Printing	0.9925	0.9925
Total	0.9793	0.9793
F1	**0.9793**	

Table B.6: Precision and Recall of Test Data of Side Camera Model

C| Research Paper

(This page is left intentionally blank)

Multi-Camera Fault Detection in Fused Deposition Modelling

A.M.A. Tournoy [a], S.Kilambi [a], M. Amani [a] , Dr. J. Jovanova [a], Dr. K. Masania [a]

[a] **Delft University of Technology**
Mekelweg 5, 2628CD Delft, the Netherlands

Corresponding Author: Shanthalakshmi Kilambi,
Other Authors: Aster Tournoy,
Muhamad Amani, Dr. Jovana Jovanova,
Dr. Kunal Masania,

Institution: Delft University of Technology
Place: Faculty of Mechanical Engineering, Delft
Present Address: Mekelweg 5, 2628CD Delft, NL
Date: February 9, 2025

Abstract

Fused Deposition Modelling (FDM) is one of the most popular 3D printing technologies because of its affordability and accessibility. FDM, however, often suffers from printing errors that result in wasted time, materials and energy. To address these challenges, our paper introduces a novel fault detection system for FDM printers. This system is designed to identify a broad range of errors without interrupting the printing process. To achieve a real-time detection system, an innovative multi-camera setup is designed, integrating two side cameras and one nozzle camera. Our hypothesis is that a system including three cameras can provide a more comprehensive view and can ensure more error types to be detected. Error detection is achieved using Convolutional Neural Networks (CNNs). This is a type of machine learning that excels at image recognition and pattern detection, making it well-suited for identifying printing errors in real-time models. Two CNN models are developed to classify images into common 3D printing errors: one model for the nozzle and another for the side cameras. The nozzle camera model achieves a high validation accuracy of 97.68% with a low loss of 0.07464. The side camera model achieves comparable performance with a validation accuracy of 97.61% and validation loss of 0.1196. These well performing models are integrated into one unified system, such that a greater range of errors can be detected. The primary innovation of this work lies in this integration. This includes a logic-based priority framework to consider discrepancies between camera outputs. From this research, we learned that integrating multiple viewpoints into a logic-driven priority framework significantly improved the robustness of error classification, as many more error types could be detected in-situ and real-time. As a result, the integrated system effectively detects 12 common printing errors. This work shows the feasibility of developing a robust multi-input fault detection system to improve 3D printing.

Key Words:
fused deposition modelling, fault detection, convolutional neural networks, multi-camera, integration

1| Introduction

Additive manufacturing, commonly known as 3D printing, is an innovative technology that rapidly and precisely transforms computer-aided design (CAD) files into physical structures [1, 2]. These structures are created by systematically joining materials layer-by-layer, using substances such as resins, inks and powders [3]. By adding material to form structures, custom design and complex structures can be fabricated while significantly reducing material waste compared to conventional manufacturing methods [4]. Among the various additive manufacturing techniques, Fused Deposition Modelling (FDM) is the most commonly used additive manufacturing technique due to its compatibility with widely available materials and ease of use [3–5]. In FDM, a continuous thermoplastic filament is unwound and liquefied to a semi-liquid state using high temperatures in the nozzle. The liquified material is then extruded under pressure [4, 6–8]. The extruded filament deposits lines onto previously made layers, which solidify upon cooling to form bottom-up 3D structures [9–11]. This layer-by-layer approach offers several advantages, including cost-effectiveness, user-friendliness and minimal post-processing requirements. However, drawbacks include relatively low resolution (100–200 [µm]) [12], low surface quality and a high frequency of defects [3, 13].

Defects in printed components occur frequently and limit the potential and benefits of FDM manufacturing [1, 14–16]. These FDM defects are particularly prevalent among inexperienced users, who encounter a failure rate of around 20% [17–20]. This high failure rate can be attributed to the open-loop nature of 3D printing. To address this, a closed-loop real-time control for FDM printers is required. One key step to achieve this is creating a fault detection system. Such a system is illustrated in Figure 1.1, where sensors are used to identify printing errors in real time. This detection system can then serve as the foundation for developing a correction mechanism, ultimately enabling a closed-loop printing process.

Several methodologies have been studied to detect and correct common mistakes in FDM. These methodologies consist of analysing image data [1, 15–17, 19–25], temperature [18, 26], acoustic emissions [27–30] and electrical resistance [14]. Among these, image data analysis has proven to be particularly effective, with studies either using a single or multi-camera setup [19, 31]. These camera-based approaches are adopted in various studies. For example, Jin et al. [22, 23] employed deep learning with Convolutional Neural Networks (CNN) to detect and correct under extrusion and over extrusion errors. Similarly, Jin et al. [21], Brion et al. [24] focused on using CNN models to identify and limit interlayer imperfections such as warping and delamination. Brion and Pattinson [15] employs image classification to categorise four printing parameters, i.e. flow rate, lateral speed, z-offset, and hotend temperature, into low, good, and high. This classification is then utilised for in-situ correction of these parameters. In another study, Petsiuk and Pearce [17] explored the use of a single-camera system to project top and side views for height validation, trajectory matching and texture analysis. Additionally, Petsiuk and Pearce [19], Nuchitprasitchai et al. [20] used histogram differences and image comparison techniques to identify defects by comparing real-time images to 2D or 3D render references.

Despite these advancements, several gaps remain in the current methodologies. First, many studies only focus on specific defect types, such as solely focussing on delamination, warping, or extrusion errors [21, 22] and without offering integrated feedback systems to detect and correct multiple types of errors in real-time. This narrow focus limits the overall effectiveness of these systems. Secondly, for many error types, there is still no dedicated detection system available. Thirdly, many existing detection systems are not in-situ, causing delays in the printing process. For example, the work by Petsiuk

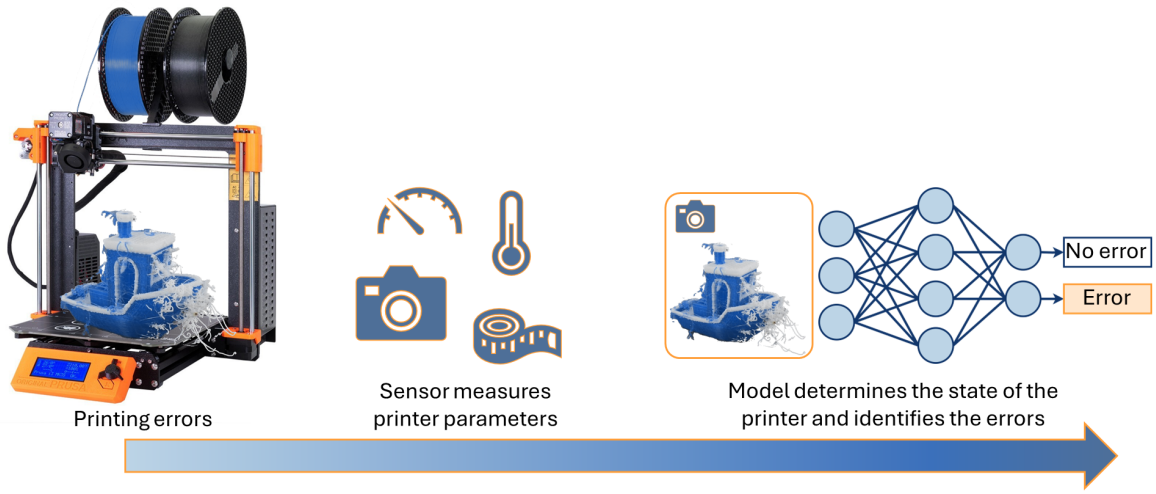


Figure 1.1: Fault detection system allows for monitoring the printer. Sensors are installed such that models can use the measurements to define and identify printing errors (adapted from [32, 33])

and Pearce [17] covers a comprehensive range of error types but is computationally inefficient, adding approximately 50% extra time to the printing process and lacking a correction procedure. Nuchitprasitchai et al. [20] faced limitations due to the time required for analysis, restricting real-time monitoring capabilities. These delays are particularly evident in methods that rely on calculating histogram differences between measured and synthetic reference images. This process is too time-consuming and unsuitable for complex geometric shapes or varying lighting conditions [19, 20]. However, using a CNN, real-time detection of errors becomes feasible, addressing the time-consuming and computational inefficiencies of traditional methods. Hence the most promising techniques involve these CNNs. Additionally, static camera setups often miss errors occurring in different parts of the print or under varying lighting conditions [24]. Single-camera systems often lack the ability to capture errors on the backside of the print or during rapid printing processes. Therefore, literature suggests that current approaches should be expanded to incorporate multi-camera setups and possibly integrate other sensor data for more comprehensive in-situ error detection and correction [15, 16, 21, 22].

Due to the current literature gaps and suggestions, this study aims to develop an integrated fault detection system for FDM printers utilising a multi-camera setup and CNN models. The primary goal is to design a rapid and highly accurate detection system capable of identifying a wide range of common printing errors real-time and in-situ. The findings demonstrate the effectiveness of the proposed system, as the CNN models developed for the multi-camera setup achieve high validation accuracies (97.5%) in detecting 12 different error types. By integrating two image classification models, robust error detection is ensured, even in complex scenarios. This research aims to lay the groundwork for future developments, including the integration of object detection methods and corrective mechanisms. This ultimately advances the reliability and efficiency of FDM printing.

2| Data Collection

2.1. Experimental Setup

The data collection system comprises three cameras: one nozzle camera and two bed cameras. The nozzle camera is mounted on the printer's heat block using a custom-designed bracket for close-up extrusion monitoring. This camera is part of the 3DO nozzle camera kit [34]. The bed cameras detect errors on the sides of the print. These cameras are attached to the printer bed frame, using brackets adapted from [35]. The bed cameras are Raspberry Pi HQ cameras [36], equipped with CS Mount 6mm 3MP-HD lenses [37]. The three cameras capture real-time images, which are transmitted to a hard drive, a Seagate Expansion Portable 2TB hard drive [38]. To manage data collection and printer control, two Raspberry Pi units are implemented: a Raspberry Pi 3 [39] running OctoPrint [40] for remote control and a Raspberry Pi 5 [41] dedicated to storing images.

This multi-camera setup is applied to a Prusa MK3S printer [42], utilising black glass fibre-reinforced polypropylene (PP GF30) [43] as material. The full configuration is illustrated schematically in Figure 2.1A, with the physical setup depicted in 2.1B.

2.2. Data Acquisition

Images of relevant and visible printing errors were collected for each camera, covering twelve distinct error types. These errors were categorised into groups based on their nature and root causes. Extrusion errors, including (i) over extrusion and (ii) under extrusion, result from incorrect material flow, leading to uneven surfaces or gaps in the print. Detail issues comprise (iii) pillowing, caused by inadequate cooling of the top layer resulting in uneven bumps, and (iv) foreign bodies, which refer to contaminants on the print bed. Structural errors involve (v) weak infill, where internal structures lack strength and (vi) poor bridging, which occurs when unsupported spans break. Temperature-related errors include (vii) stringing, characterised by thin strands of filament between parts of the print, and (viii) overhang sagging, where unsupported regions droop. Alignment and motion errors include (ix) shifted layers, caused by mechanical misalignments and (x) spaghetti failure, characterised by chaotic filament strands. Lastly, adhesion errors, such as (xi) warping and (xii) delamination, are caused by insufficient layer bonding, leading to parts lifting from the bed or separating between layers. Table 2.1 summarises these errors and their corresponding detecting cameras. [44–46]

Error group	Error	Detecting Camera
Extrusion Errors	Over Extrusion	Nozzle Camera
	Under Extrusion	Nozzle Camera
Detail Issues	Pillowing	Nozzle Camera
	Foreign Body in Print	Side Cameras
Structural Errors	Weak Infill	Nozzle Camera
	Poor Bridging	Side Cameras
Temperature Errors	Stringing	Nozzle and Side Cameras
	Overhang Sagging	Side Cameras
Alignment and Motion Errors	Shifted Layers	Side Cameras
	Spaghetti Failure	Nozzle and Side Cameras
Adhesion Errors	Warping	Side Cameras
	Delamination	Side Cameras

Table 2.1: Included error types and their detecting cameras

Three errors were found to trigger other detectable faults rather than presenting distinct visual features: (i) support falling apart, (ii) bed separation and (iii) overheating. These three error types were categorised under their visible outcomes. The former two resulted in spaghetti failure, while the latter led to stringing, overhang sagging, poor bridging and warping.

Seven other errors were excluded from the detection system: (i) small features not printing, (ii) poor surface quality at support regions, (iii) missing layers, (iv) first layer problems, (v) ripples and echoes, (vi) infill showing on the exterior of the print and (vii) elephant foot. The first four errors are omitted as they are not easily visible without a comparison to external references, such as CAD files. Ripples and echoes require vibration testing equipment, which was beyond the study's scope. Lastly, infill showing on the exterior and elephant foot were excluded because their subtle nature made detection unreliable.

A minimum of 2000 images per error type were collected for the side camera and 2400 per error type for the nozzle camera, ensuring variability in shapes, infill patterns and conditions in the images. The images were labelled according to their respective error classes and subsequently balanced so all classes contained an equal number of images (2000 images for the side camera and 2400 for the nozzle camera). The datasets were then split into training, validation and test sets using a 70-20-10 ratio. To enhance the model's generalisation capability, data augmentation techniques were applied to the training data. These techniques involved transforming the images while maintaining their original labels. Specific transformations included horizontal flipping, rotations and colour adjustments, such as altering brightness, contrast and saturation. The labelled and augmented images are shown in Figure 2.1C. Additionally, an external dataset, featuring a distinct shape, was created to evaluate the models' ability to detect errors in previously unseen data.

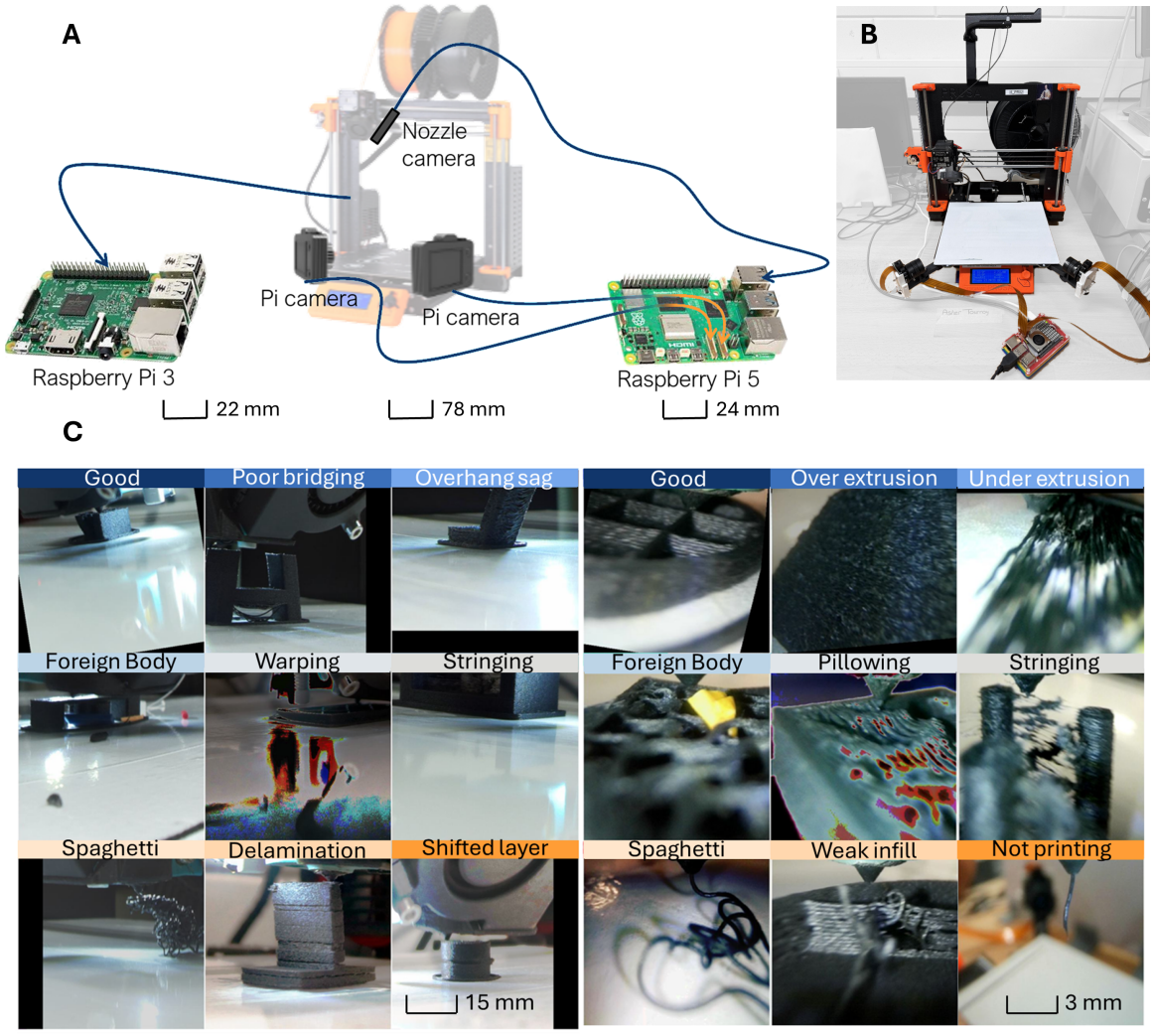


Figure 2.1: Data collection. A: The graphical setup of data collection consisting of three cameras and two raspberry pis (adapted from [32, 47–49]). B: The physical setup. C: Data labelling and augmentation, left side shows the augmented images of the side camera, the right side shows the augmented images of the nozzle camera.

3| Results and Discussion

3.1. Model Results and Discussion

3.1.1. Nozzle Camera Model

The nozzle camera model is trained for 10 epochs using a Convolutional Neural Network (CNN) architecture with the YOLOv8 classification framework. This YOLOv8 framework is known for its efficiency in real-time object detection and classification [50], thus offering a robust pre-trained model. Our nozzle camera model achieved a validation accuracy of 97.68% and a test accuracy of 97.09%, demonstrating its strong performance in classifying printing errors. These results of the the training, validation, test and external test datasets are summarised in Table 3.1, including the accuracy, softmax cross-entropy loss and the F1 scores. High accuracy, low loss and high F1 score confirm the model's ability to capture patterns effectively. This is also confirmed using the confusion matrices of the four datasets, depicted in Figure 3.1A. In these matrices, dark blue diagonal boxes indicate correct classifications and orange boxes represent misclassifications. These matrices illustrate high classification performance in training validation and test sets across different classes.

Metric	Training	Validation	Test	External Test
Accuracy (%)	99.11	97.68	97.09	93.18
Loss	0.07153	0.07464	0.07478	0.2770
F1 score (%)	99.11	97.66	97.10	91.09

Table 3.1: Performance of nozzle camera model: the accuracy, loss and F1 score of the training, validation, test and external test sets

However, some challenges were identified in the nozzle camera model. Using the external validation dataset, the loss increased to 0.2270 and accuracy dropped to 93.18%. This small decrease of 4% is likely due to domain-specific changes such as shape variations and misclassifications of specific classes. The over extrusion class had a decrease in recall, reflecting false negatives. This issue is potentially due to overlapping features or insufficient sample diversity in the training set. The stringing class showed a decrease in precision, with false positives often confused with weak infill defects. Performance of the weak infill class was also lower, though the small external dataset of weak infill limits the ability to draw conclusions.

Compared to prior studies, the nozzle camera model demonstrates competitive performance. For instance, Jin et al. [21] reported a test accuracy of 91.0% for nozzle height classification, while another study by Jin et al. [23] achieved 98% accuracy for extrusion fault classification, including successful in-situ corrections. These studies used white polylactic acid (PLA), a material and colour that is typically easier to process and provides higher contrast, unlike black PP GF30 filament used in this study. Additionally, our study introduced an external validation set, providing stronger evidence for the model's generalisability across different conditions. Strengths of our model include high accuracy, generalisability and balanced classification across most fault categories. However, weaknesses in handling over extrusion and stringing faults suggest the need for more diverse training data to improve robustness.

3.1.2. Side Camera Model

The side camera model utilised the pre-trained YOLOv8 model for 30 epochs. Our side camera model achieved high accuracy, low loss and high F1 scores across training, validation and test datasets, demonstrating reliable classification performance. These results, presented in Table 3.2, are further supported by the confusion matrices for the datasets illustrated in Figure 3.1B. The confusion matrices for the training, validation and test datasets revealed minimal misclassifications across the 11 classes.

Metric	Training	Validation	Test	External Test
Accuracy (%)	99.81	97.61	97.86	89.30
Loss	0.04255	0.1196	0.089906	0.3927
F1 score (%)	99.82	97.64	97.93	84.06

Table 3.2: Performance of side camera model: the accuracy, loss and F1 score of the training, validation, test and external test sets

When applying the side camera model to the external dataset, the model faced adaptation challenges. These challenges were particularly evident in classes such as stringing, poor bridging, and shifted layers, which were frequently misclassified as warping. However, many of those misclassified images frequently contained multiple co-occurring defects. This suggests that the model's actual performance in detecting faults may be underestimated. Additionally, the precision for the overhang sag class was relatively low, indicating false positives. This limitation could be attributed to the shape of the external test, which has dominant overhanging features, that may have confused the model.

With a validation accuracy of 97.61%, the side camera model demonstrates competitive performance compared to existing studies. Models in these studies, utilising bed cameras, reported accuracies ranging from 83% to 99.3% for specific fault detection tasks such as warping detection or spaghetti detection [1, 24, 25, 51, 52]. Despite handling 11 classes, our model achieved high accuracies, highlighting its robustness. However, its performance on the external dataset reveals opportunities for improvement. For instance, a more diverse training dataset could enhance its generalisability, particularly for conditions outside the training domain. Furthermore, addressing challenges such as overlapping defect features and adapting to varying print characteristics is essential to further improve the model's reliability in real-world applications.

3.1.3. Literature Overview

Table 3.3 provides a comparative analysis of the developed models against existing fault detection systems. This table lists, per source, the type of printer, the type of sensors used, the classes of the models and their accuracies. This comparison is further visualised in Figure 3.1C. This plot shows the amount of error types existing literature cover and their validation accuracy. It can be noted that our system covers many more different error types while achieving similar or even higher accuracies, highlighting its suitability for the integrated fault detection system.

Source	Printer	Sensor	Classes	Acc. (%)
[21]	FDM	Nozzle camera	nozzle height: high +, high, good, low	91.0
[23]	FDM	Nozzle camera	good, over, under extrusion	98.0
[15]	FDM	Nozzle camera	flow rate, speed, temp, Z-offset: low, good, high	90.2
[1]	FDM	Bed camera	normal, stringing, under extrusion	83
[24]	FDM	Bed camera	warping detection	88.72
[25]	FDM	Bed camera	warping detection	99.3
[51]	FDM	Bed camera	stringy, obstacle, unstick	89.7
[53]	FDM	Bed camera	acceptable, faulty	70.0
[54]	FDM	Bed camera	Print speed value, temperature value	94.0
[52]	FDM	Bed camera	spaghetti object detection	96.0
[55]	FDM	Rotating bed camera	normal, blob, void, thick line, crack, misalignment	91.0
[56]	FDM	Top-view camera	acceptable, faulty	84.95
[57]	VAT	Temperature, vibration, humidity	acceptable, faulty	80
[58]	L-PBF	Photodiode	acceptable, faulty	77
[59]	L-PBF	Thermographic camera	ok, metal splatters and delamination	96.8
own	FDM	Nozzle and 2 bed cameras	15 classes , containing 12 errors	97.7, 97.6

Table 3.3: Comparison of fault detection methods using NN in literature in additive manufacturing

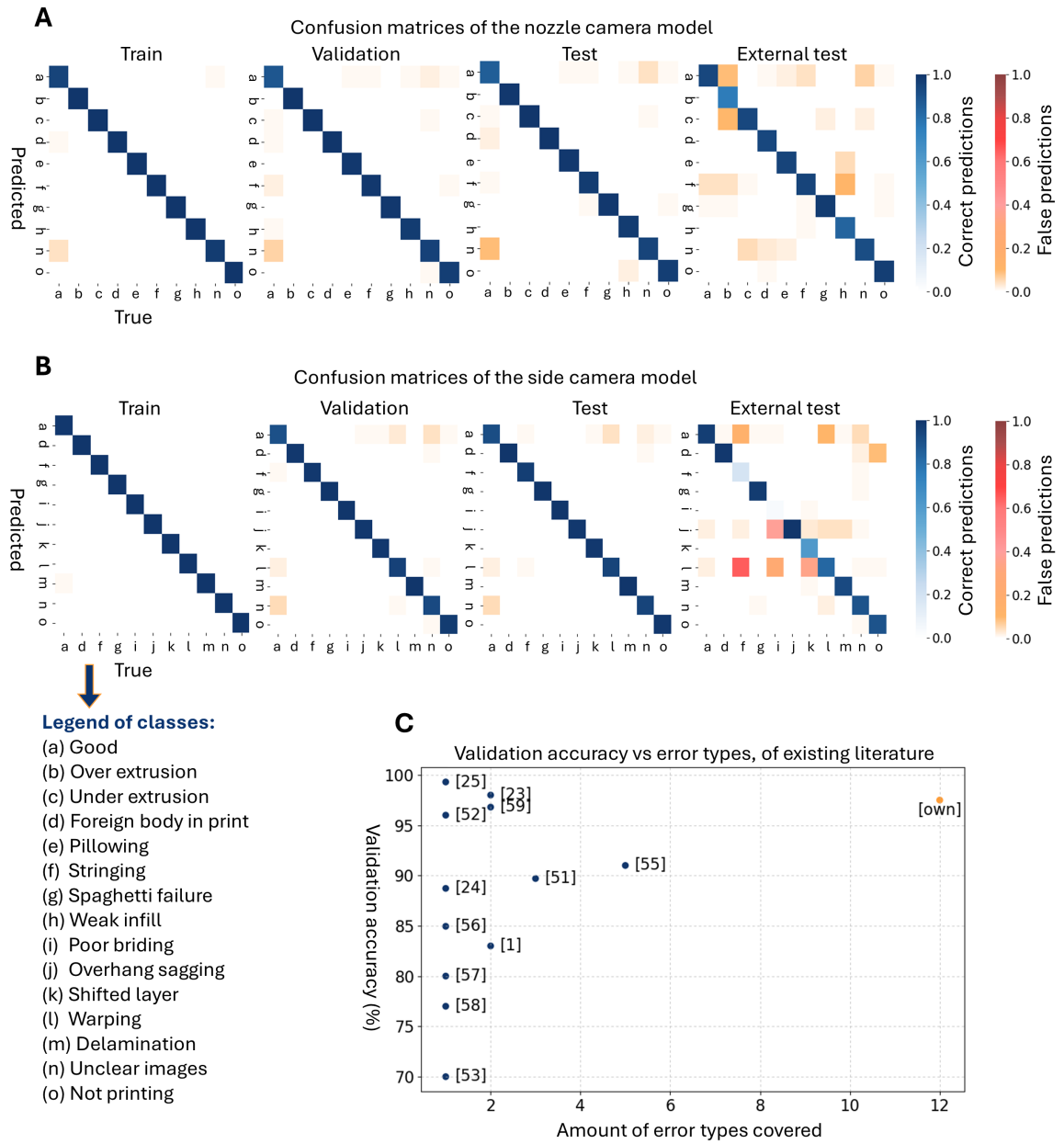


Figure 3.1: Results of the models. A: The confusion matrices of the nozzle camera model, for the training, validation, test and external test set, the individual classes are labelled according to the legend. B: The confusion matrices of the side camera model, for the training, validation, test and external test set. C: The validation accuracy of our system compared to existing literature, considering the amount of error types covered.

3.2. Fault Detection System

3.2.1. Outline of Detection System

The detection system operates in a continuous loop, capturing three images during each iteration: one from the nozzle camera and two from the side cameras (left and right). These images are processed separately, resulting in three outputs that are then integrated into a single conclusion. This innovative process is visualised in Figure 3.2A. Before being passed through the models, the images from the nozzle and side cameras are cropped. Cropping ensured the images better resembled the training data. Training images were cropped to prevent the models from exceeding memory limitations, as the original image sizes were too large to process effectively. Additionally, cropping helped eliminate unnecessary noise, allowing the analysis to focus on the relevant print areas. These areas differ for the nozzle camera and side cameras.

The nozzle camera provides a fixed perspective due to the stationary position of the nozzle relative to the camera. This consistency allows for uniform cropping of images, reducing the input size from 640x480 to 440x320 pixels. These cropped images are then analysed by the dedicated CNN model to detect errors specific to this perspective. For the side cameras, cropping requires manual adjustments. This is because new prints can be positioned differently on the bed, resulting in no fixed reference point for the camera. During the first loop, the operator selects a 1600x1000-pixel region (from images of 4056x3040 pixels) focusing on the print area. This cropping region is fixed for all following iterations, as the position of the print does not move with respect to the side cameras, during the same print. The cropped side images are analysed by the respective CNN model, tailored to detect errors observable from the side perspectives, resulting in the classification outputs. The outputs from the three cameras are then passed to the logic system for integration, ensuring prioritised and accurate error reporting.

3.2.2. System Logic

The logic system transforms raw model outputs into actionable conclusions. First the individual answers for each camera are derived, based on consecutive outputs of that camera. Secondly, these answers are combined into a unified system-level conclusion. The two-step process ensures consistency and prioritisation. This is presented in the right part of Figure 3.2A.

Each camera's outputs are saved in separate lists. The output lists undergo a filtering process to addresses cases where outputs such as "good", "unclear" or "not printing" appear. These require special handling to avoid disrupting error detection. First, if a model predicts "good," this output is tracked and a counter is incremented for consecutive "good" predictions. Once four consecutive "good" predictions are observed, the list of previous outputs is cleared. This clearing ensures that, for example, if there are 1500 predictions, with 1496 classified as good and only 4 scattered as stringing, the system avoids incorrectly returning stringing. Secondly, "unclear" outputs are temporarily included for evaluation but subsequently removed. As a consequence, "unclear" images do not disturb the sequence. Thirdly, outputs "not printing" are excluded unless detected simultaneously by two or more cameras, as a single "not printing" prediction may result from nozzle movements rather than an actual fault.

Error confirmation relies on consistency across iterations. If an output, such as "stringing", is repeated consistently across four or more iterations for one camera, it is finalised as the camera's answer. This is also the case when two cameras detect the same error simultaneously twice in a row. Similarly, when all three cameras agree on the same error, this answer will be returned. In all three cases, the system tolerates an intermittent "good" or "unclear" predictions within the sequence to account for model inaccuracies. This logic is statistically robust, ensuring a minimum detection accuracy of 99% even when individual model accuracies are approximately 97%.

The final step in the fault detection system integrates the outputs from the three cameras into a single cohesive conclusion. This requires an integration logic that prioritises errors based on context. For instance, when side cameras detect overhang sag (OS), poor bridging (PB) or delamination (DE), the nozzle camera's output is ignored. If the side cameras do not detect any of these three errors and the nozzle camera identifies over extrusion (OE) or under extrusion (UE), the side camera outputs are disregarded for that iteration. Disregarding cameras in these scenarios ensures that errors detectable by one camera, but not by others, do not introduce incorrect conclusions due to overlapping features. So, by prioritising and ignoring specific camera outputs, the system avoids conflicting conclusions.

This integration ensures a robust and responsive detection system. The system provides accurate error identification while minimising disruptions caused by transient or isolated mispredictions. An example of the system's operation over four iterations is shown in Figure 3.2B.

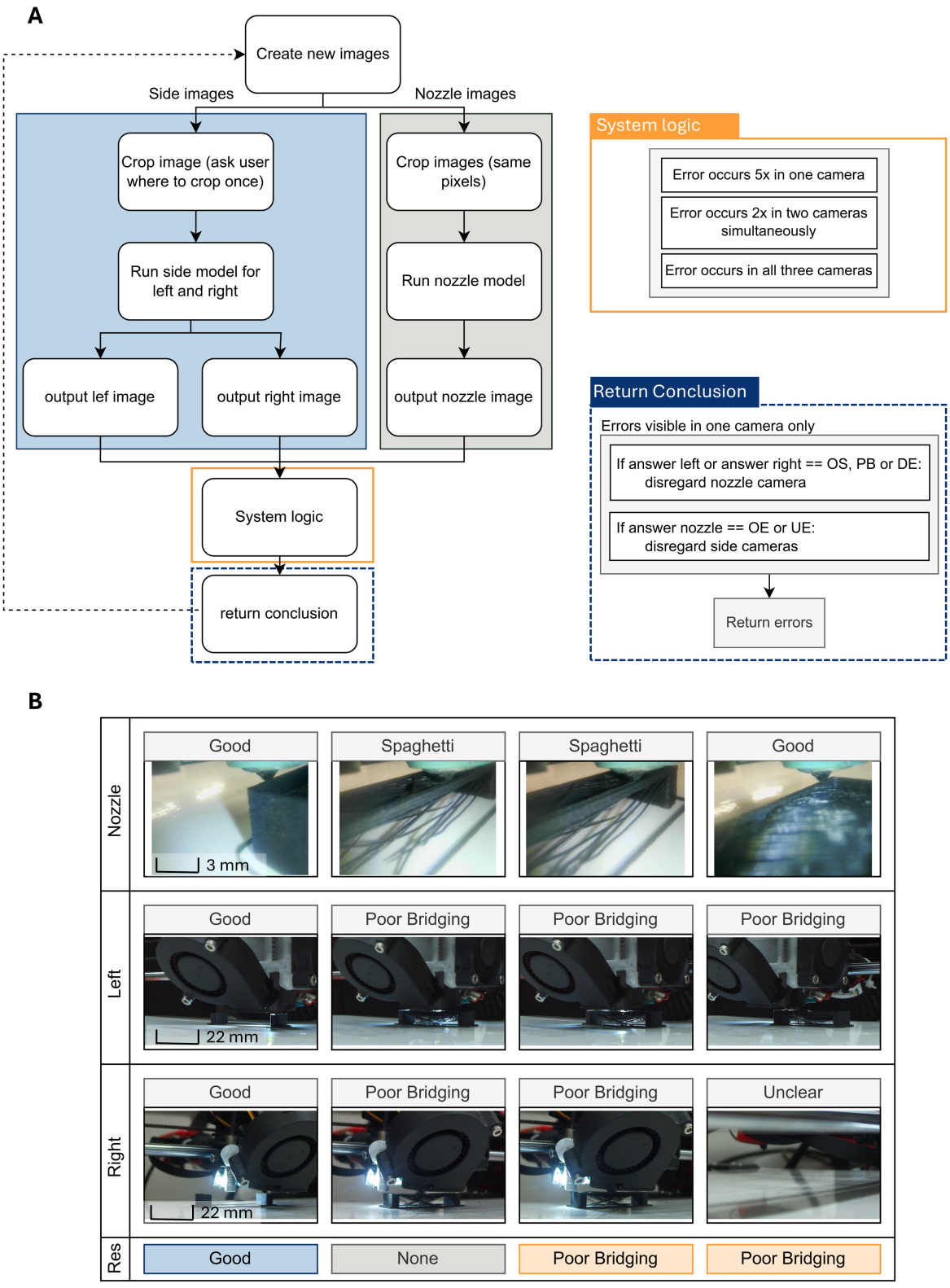


Figure 3.2: Fault detection system. A: Outline of the system, comprising of a cropping mechanism, applying the models, retrieving the outputs, applying a logic to those outputs to form a unified conclusion. B: Example case of the detection mechanism, where twice in a row two cameras detected the same error: poor bridging, hence that becomes the conclusion.

4| Further Scope

4.1. Fault Correction System

Looking forward, the system could be further developed to include real-time adjustments to printer settings. This creates a comprehensive error correction mechanism and makes the printing process closed loop, as illustrated in Figure 4.1A. This could be achieved by modifying extrusion rates for over and under extrusion issues, adjusting cooling for temperature-related problems and more. Modifying printing parameters real-time would pave the way for fully automated, self-regulating and reliable 3D printing processes.

Beyond classification-based detection, object detection can enhance fault identification by pinpointing the exact regions of defects within the print. This would allow for precise, targeted corrections. This would also enable the system to recognise multiple errors occurring simultaneously. An example of object detection for 3D printing fault detection is shown in Figure 4.1B, demonstrating how multiple errors can be identified and their locations mapped in real-time.

Additionally, severity estimation should be incorporated to distinguish between minor and critical faults. This could be achieved using confidence scoring within the object detection framework. Creating intervention priorities could prevent unnecessary print interruptions while ensuring major defects are addressed promptly. This would create a more intelligent, dynamic and responsive fault detection system.

4.2. Other Applications

The fault detection system developed holds significant potential for future applications in additive manufacturing. The scope can be extended due to the adaptability of the current methodology. It can be adapted for diverse printing configurations and scales. Two key applications include 45° belt printing and large-scale printing.

One promising application of the system is 45° belt printing. This method offers advantages such as continuous production and the ability to print long objects [60–62]. To achieve a fault detection system for 45° printing, the system would require repositioning cameras and adjusting detection algorithms. This is required to account for the unique printing angle and the unique errors associated with this method. Including such adaptations could enhance the efficiency and reliability of 45° belt printing processes, making it a viable solution for industrial applications.

For large-scale 3D printing applications, the fault detection system holds even greater promise. This is due to the significantly larger print volumes and longer print times involved in large-scale printing, which make error detection even more crucial. The detection system could be adapted and integrated into print-robot setups, such as the ABB6700 robot [63] mounted on a rail system used by Tanaruz [64]. As illustrated in Figure 4.1C, bed cameras can be strategically positioned on a beam mounted on the rails to monitor the entire printing area. This setup is complemented by a nozzle camera on the robot arm, which provides real-time material deposition feedback. Monitoring the large-scale nozzles does no longer require specialised close-up equipment like borescopes. That is because of the larger nozzle size and greater distance between the nozzle and camera. This could reduce costs and increase reliability for industrial applications.

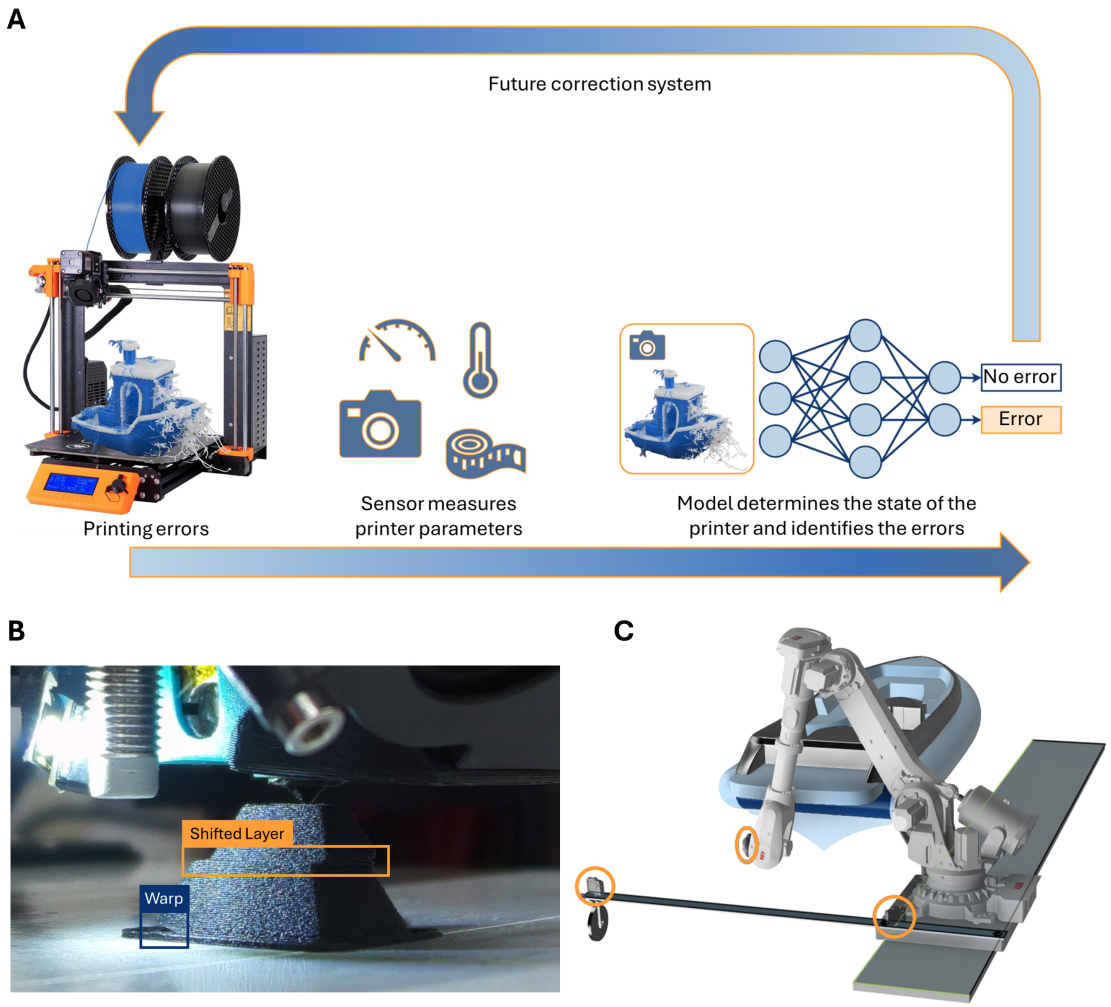


Figure 4.1: Future scope of the fault detection system. A: Making the process closed loop by integrating a correction mechanism. B: Object detection system can detect two errors simultaneously and locate errors, which allows for precise correction. C: The scope of the fault detection system can be extended to other FDM methods, such as large-scale FDM printing (adapted from [47, 65, 66])

5| Conclusion

This study developed a multi-camera fault detection system for FDM 3D printers, comprising one nozzle camera and two bed cameras. The multi-camera system successfully integrated two CNN models, one for the nozzle camera and one for the bed cameras, into a unified framework. This framework reliably detected 12 common error types across different perspectives. These errors include over extrusion, under extrusion, pillowing, foreign bodies in the print, weak infill, poor bridging, stringing, overhang sagging, shifted layers, spaghetti failure, warping and delamination.

The nozzle camera model achieved a validation accuracy of 97.68% and the side camera model performed comparably with a validation accuracy of 97.61%. While these results highlight the system's strong performance on controlled datasets, challenges emerged during testing on the external validation set. The accuracy for this external set dropped to 93.18% for the nozzle camera and 89.3% for the side cameras. These reductions can be partially attributed to the shape of the external test set and the co-occurrence of errors. Due to these reasons, the external dataset performance might underestimate the system's true performance. However, there still remains a considerable drop in performance which should be addressed. Addressing these limitations could involve expanding the dataset to encompass a wider variety of conditions and incorporating domain adaptation techniques to improve robustness.

The multi-camera integration logic ensured robust error classification by prioritising outputs based on error context. Prioritisation is achieved through a logic-based framework that resolved discrepancies between camera outputs, ensuring consistent and reliable error detection. The detection is further enhanced by relying on repeated error outputs from the models. This results in a detection accuracy exceeding 99%. The high level detection mechanism supports real-time monitoring without disrupting the printing process. As a result, the system significantly contributes to reduced material waste and improved print reliability, making it a valuable tool for both industrial and desktop 3D printing applications.

This research demonstrates the feasibility of combining machine learning models into a comprehensive in-situ fault detection system. The system's performance could be enhanced using future developments. These developments include expanding the dataset with more materials, colours, shapes, printers and lighting conditions. Potential future work consists of creating an object detection model that also includes the severity of the errors, rather than an error-type classification system. Then, research could be expanded to include real-time error correction, paving the way for fully automated and efficient 3D printing workflows.

Data Availability

- The python scripts of the CNN models and integrated fault detection system is available at 10.17632/nm2pytfw8f.1 [NOT YET PUBLISHED]
- The image data that support the findings of this study are available from the corresponding author, A.M.A. Tournoy, upon request.

CRediT

- A.M.A. Tournoy conceptualisation, methodology, software, formal analysis, investigation, resources, data curation, writing - original draft, visualisation
- S. Kilambi: methodology, validation, resources, writing - review & editing, supervision
- M. Amani: conceptualisation, supervision
- Dr. J. Jovanova: conceptualisation, writing - review & editing, supervision, project administration
- Dr. K. Masania: conceptualisation, resources, writing - review & editing, supervision

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used Open AI ChatGPT 4.0 to check for spelling and grammar mistakes and improve readability. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Acknowledgement

We would like to express our sincere gratitude to Delft University of Technology and the Faculty of Mechanical Engineering for their invaluable support throughout this research. Our heartfelt thanks go to the Shaping Matter Lab and Tanaruz Boats for generously providing the tools and infrastructure that made this work possible. We are especially grateful to Kurt Tournoy and Ulf Kemmsies for their thoughtful feedback and careful review of this paper. We also wish to thank Carolina Martellotto for her dedication and assistance with data annotation. Finally, we appreciate the support and encouragement from our friends and family, which has been a constant source of motivation throughout this work.

References

[1] Luca Rettenberger, Nils Beyer, Ingo Sieber, and Markus Reischl. Fault detection in 3d-printing with deep learning. In *2024 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–4, 2024. doi: 10.1109/ICCE59016.2024.10444198.

[2] Zeqing Jin, Zhizhou Zhang, Kahraman Demir, and Grace X. Gu. Machine learning for advanced additive manufacturing. *Matter*, 3(5): 1541–1556, 2020. ISSN 2590-2385. doi: https://doi.org/10.1016/j.matt.2020.08.023.

[3] Jahan Zeb Gul, Memoon Sajid, Muhammad Muqeet Rehman, Ghayas Uddin Siddiqui, Imran Shah, Kyung-Hwan Kim, Jae-Wook Lee, and Kyung Hyun Choi. 3d printing for soft robotics - a review. *Science and Technology of Advanced Materials*, 19(1):243–262, 2018. doi: 10.1080/14686996.2018.1431862. PMID: 29707065.

[4] Jigar Patadiya, Adwait Gawande, Ganapati Joshi, and Balasubramanian Kandasubramanian. Additive manufacturing of shape memory polymer composites for futuristic technology. *Industrial & Engineering Chemistry Research*, 60, 10 2021. doi: 10.1021/acs.iecr.1c03083.

[5] Aamir Ahmed, Sandeep Arya, Vinay Gupta, Hidemitsu Furukawa, and Ajit Khosla. 4d printing: Fundamentals, materials, applications and challenges. *Polymer*, 228:123926, 2021. ISSN 0032-3861. doi: https://doi.org/10.1016/j.polymer.2021.123926.

[6] Ajit Behera. Smart materials based additive manufacturing. In Shashanka Rajendrachari, editor, *Practical Implementations of Additive Manufacturing Technologies*, chapter 7, pages 153–175. Springer Nature, Singapore, 2024. doi: 10.1007/978-981-99-5949-5_7.

[7] Muhammad Yasir Khalid, Zia Ullah Arif, Ali Tariq, Mokarram Hossain, Kamran Ahmed Khan, and Rehan Umer. 3d printing of magneto-active smart materials for advanced actuators and soft robotics applications. *European Polymer Journal*, 205:112718, 2024. ISSN 0014-3057. doi: https://doi.org/10.1016/j.eurpolymj.2023.112718.

[8] Ryan T. Shafranek, S. Cem Millik, Patrick T. Smith, Chang-Uk Lee, Andrew J. Boydston, and Alshakim Nelson. Stimuli-responsive materials in additive manufacturing. *Progress in Polymer Science*, 93:36–67, 2019. ISSN 0079-6700. doi: https://doi.org/10.1016/j.progpolymsci.2019.03.002.

[9] Sophie Leanza, Shuai Wu, Xiaohao Sun, H. Jerry Qi, and Ruike Renee Zhao. Active materials for functional origami. *Advanced Materials*, 36(9), December 2023. ISSN 1521-4095. doi: 10.1002/adma.202302066.

[10] Xueli Zhou, Luquan Ren, Zhengyi Song, Guiwei Li, Jifeng Zhang, Bingqian Li, Qian Wu, Wangxuan Li, Lei Ren, and Qingping Liu. Advances in 3d/4d printing of mechanical metamaterials: From manufacturing to applications. *Composites Part B: Engineering*, 254:110585, 2023. ISSN 1359-8368. doi: https://doi.org/10.1016/j.compositesb.2023.110585.

[11] Prasansha Rastogi and Balasubramanian Kandasubramanian. Breakthrough in the printing tactics for stimuli-responsive materials: 4d printing. *Chemical Engineering Journal*, 366:264 – 304, 2019. doi: 10.1016/j.cej.2019.02.085. Cited by: 191.

[12] Amal Megdich, Habibi Mohamed, and Luc Laperrière. A review on 4d printing: Material structures, stimuli and additive manufacturing techniques. *Materials Letters*, 337:133977, 02 2023. doi: 10.1016/j.matlet.2023.133977.

[13] Donghwan Song, Adrian Matias Chung Baek, Jageon Koo, Moise Busogi, and Namhun Kim. Forecasting warping deformation using multivariate thermal time series and k-nearest neighbors in fused deposition modeling. *Applied Sciences*, 10(24), 2020. ISSN 2076-3417. doi: 10.3390/app10248951.

[14] Heime Jonkers, Alexander Dijkshoorn, Stefano Stramigioli, and Gijs Krijnen. Fff print defect characterization through in-situ electrical resistance monitoring. *Scientific Reports*, 14(1):11906, May 2024. ISSN 2045-2322. doi: 10.1038/s41598-024-59053-z.

[15] Douglas A. J. Brion and Sebastian W. Pattinson. Generalisable 3d printing error detection and correction via multi-head neural networks. *Nature Communications*, 13(1):4654, Aug 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-31985-y.

[16] Douglas A. J. Brion and Sebastian W. Pattinson. Quantitative and real□time control of 3d printing material flow through deep learning. *Advanced Intelligent Systems*, 4(11), September 2022. ISSN 2640-4567. doi: 10.1002/aisy.202200153. URL http://dx.doi.org/10.1002/aisy.202200153.

[17] Aliaksei L. Petsiuk and Joshua M. Pearce. Open source computer vision-based layer-wise 3d printing analysis. *Additive Manufacturing*, 36: 101473, 2020. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2020.101473.

[18] B.T. Wittbrodt, A.G. Glover, J. Laureto, G.C. Anzalone, D. Oppliger, J.L. Irwin, and J.M. Pearce. Life-cycle economic analysis of distributed manufacturing with open-source 3-d printers. *Mechatronics*, 23(6):713–726, 2013. ISSN 0957-4158. doi: https://doi.org/10.1016/j.mechatronics.2013.06.002.

[19] Aliaksei Petsiuk and Joshua M. Pearce. Towards smart monitored am: Open source in-situ layer-wise 3d printing image anomaly detection using histograms of oriented gradients and a physics-based rendering engine. *Additive Manufacturing*, 52, 2022. doi: 10.1016/j.addma.2022.102690.

[20] Siranee Nuchitprasitchai, Michael Roggemann, and Joshua M. Pearce. Factors effecting real-time optical monitoring of fused filament 3d printing. *Progress in Additive Manufacturing*, 2(3):133–149, Sep 2017. ISSN 2363-9520. doi: 10.1007/s40964-017-0027-x.

[21] Zeqing Jin, Zhizhou Zhang, and Grace X. Gu. Automated real□time detection and prediction of interlayer imperfections in additive manufacturing processes using artificial intelligence. *Advanced Intelligent Systems*, 2(1), December 2019. ISSN 2640-4567. doi: 10.1002/aisy.201900130.

[22] Zeqing Jin, Zhizhou Zhang, Joshua Ott, and Grace X. Gu. Precise localization and semantic segmentation detection of printing conditions in fused filament fabrication technologies using machine learning. *Additive Manufacturing*, 37:101696, 2021. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2020.101696.

[23] Zeqing Jin, Zhizhou Zhang, and Grace X. Gu. Autonomous in-situ correction of fused deposition modeling printers using computer vision and deep learning. *Manufacturing Letters*, 22:11–15, 2019. ISSN 2213-8463. doi: https://doi.org/10.1016/j.mfglet.2019.09.005.

[24] Douglas A.J. Brion, Matthew Shen, and Sebastian W. Pattinson. Automated recognition and correction of warp deformation in extrusion additive manufacturing. *Additive Manufacturing*, 56:102838, 2022. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2022.102838.

[25] Aditya Saluja, Jiarui Xie, and Kazem Fayazbakhsh. A closed-loop in-process warping detection system for fused filament fabrication using convolutional neural networks. *Journal of Manufacturing Processes*, 58:407–415, 2020. ISSN 1526-6125. doi: https://doi.org/10.1016/j.jmapro.2020.08.036.

[26] E. Ferraris, J. Zhang, and B. Van Hooreweder. Thermography based in-process monitoring of fused filament fabrication of polymeric parts. *CIRP Annals*, 68(1):213–216, 2019. ISSN 0007-8506. doi: https://doi.org/10.1016/j.cirp.2019.04.123.

[27] Jie Liu, Youmin Hu, Bo Wu, and Yan Wang. An improved fault diagnosis approach for fdm process with acoustic emission. *Journal of Manufacturing Processes*, 35:570–579, 2018. ISSN 1526-6125. doi: https://doi.org/10.1016/j.jmapro.2018.08.038.

[28] *A New Approach for Online Monitoring of Additive Manufacturing Based on Acoustic Emission*, volume Volume 3: Joint MSEC-NAMRC Symposia of *International Manufacturing Science and Engineering Conference*, 06 2016. doi: 10.1115/MSEC2016-8551.

[29] Haixi Wu, Zhonghua Yu, and Yan Wang. Real-time fdm machine condition monitoring and diagnosis based on acoustic emission and hidden semi-markov model. *The International Journal of Advanced Manufacturing Technology*, 90(5):2027–2036, May 2017. ISSN 1433-3015. doi: 10.1007/s00170-016-9548-6.

[30] Pascal Becker, Christian Roth, Arne Roennau, and Ruediger Dillmann. Acoustic anomaly detection in additive manufacturing with long short-term memory neural networks. In *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*, pages 921–926, 2020. doi: 10.1109/ICIEA49774.2020.9102002.

[31] Alexander Oleff, Benjamin Küster, Malte Stonis, and Ludger Overmeyer. Process monitoring for material extrusion additive manufacturing: a state-of-the-art review. *Progress in Additive Manufacturing*, 6(4):705–730, Dec 2021. ISSN 2363-9520. doi: 10.1007/s40964-021-00192-4.

[32] 3DPrinterGear. Original prusa i3 mk3s+ assembled. https://www.3dprintergear.com.au/original-prusa-i3-mk3s-assembled, 2024. Retrieved on December 28, 2024.

[33] Beau Jackson. 3d printing industry: The 3d benchy 'torture test' that pushes 3d printers to the limit. https://3dprintingindustry.com/news/3d-benchy-torture-test-pushes-3d-printers-limit-103662/, 20197. Retrieved on January 30, 2025.

[34] 3do. Enclosure nozzle camera v2 kit. https://3do.dk/3do-kamera/2333-3do-enclosure-nozzle-camera-v2-kit.html, 2024. Retrieved on August 9, 2024.

[35] Printables. Derek_184700, octopi camera bed mount for i3mk3s+. https://www.printables.com/model/125877-octopi-camera-bed-mount-for-i3mk3s, 2024. Retrieved on December 8, 2024.

[36] Raspberry Pi. High quality camera lense 6mm 3mp wide angle. https://thepihut.com/products/raspberry-pi-high-quality-camera-lens, 2024. Retrieved on December 7, 2024.

[37] Ivy Liam Gadgets. Raspberry pi hq camera 6mm 3mp lens (cs-mount). https://shop.ivyliam.com/product/6mm-3mp-lens-cs-mount/, 2024. Retrieved on December 8, 2024.

[38] seagate. External hard drives: Protable drive. https://www.seagate.com/gb/en/products/external-hard-drives/portable-drive/, 2025. Retrieved on January 3, 2025.

[39] Raspberry Pi. Raspberry pi 3 model b. https://www.raspberrypi.com/products/raspberry-pi-3-model-b/, 2024. Retrieved on December 8, 2024.

[40] OctoPrint. The snappy web interface for your 3d printer. https://octoprint.org/, 2024. Retrieved on August 13, 2024.

[41] Raspberry Pi. Raspberry pi 5. https://www.raspberrypi.com/products/raspberry-pi-5/, 2024. Retrieved on December 8, 2024.

[42] Prusa. Prusa mk3s. https://www.prusa3d.com/product/original-prusa-i3-mk3-s-to-mk4-upgrade-kit/, 2024. Retrieved on December 5, 2024.

[43] Transmare. Cirqlair. https://transmarecirqlair.com/, 2024. Retrieved on August 13, 2024.

[44] Sourced. 44 common 3d print problems - troubleshooting issues 2023. https://www.3dsourced.com/rigid-ink/ultimate-3d-printing-troubleshooting-guide/, 2023. Retrieved on August 1, 2024.

[45] All3DP. 3d printing troubleshooting: All problems and solutions. https://all3dp.com/1/common-3d-printing-problems-troubleshooting-3d-printer-issues/, 2024. Retrieved on August 1, 2024.

[46] MatterHackers. 3d printer troubleshooting guide. https://www.matterhackers.com/articles/3d-printer-troubleshooting-guide#Issue8, 2024. Retrieved on August 1, 2024.

[47] Paul Gherghisan. Grabcad community, simple photo camera. https://grabcad.com/library/simple-photo-camera-1, May 2024. Retrieved on December 25, 2024.

[48] Electronica voor jou. Raspberry pi 5 4gb. https://elektronicavoorjou.nl/en/product/raspberry-pi-5-4gb, 2024. Retrieved on December 28, 2024.

[49] Kiwi-electronics. Raspberry pi 3 b+. https://www.kiwi-electronics.com/nl/raspberry-pi-3-model-bplus-3392, 2024. Retrieved on December 28, 2024.

[50] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. pages 779–788, 06 2016. doi: 10.1109/CVPR.2016.91.

[51] Nyoman Bogi Aditya Karna, Made Adi Paramartha Putra, Syifa Maliah Rachmawati, Mideth Abisado, and Gabriel Avelino Sampedro. Toward accurate fused deposition modeling 3d printer fault detection using improved yolov8 with hyperparameter optimization. *IEEE Access*, 11: 74251 – 74262, 2023. doi: 10.1109/ACCESS.2023.3293056. Cited by: 18; All Open Access, Gold Open Access.

[52] Hyungjung Kim, Hyunsu Lee, Ji-Soo Kim, and Sung-Hoon Ahn. Image-based failure detection for material extrusion process using a convolutional neural network. *The International Journal of Advanced Manufacturing Technology*, 111(5):1291–1302, Nov 2020. ISSN 1433-3015. doi: 10.1007/s00170-020-06201-0. URL https://doi.org/10.1007/s00170-020-06201-0.

[53] Zhicheng Zhang, Ismail Fidan, and Michael Allen. Detection of material extrusion in-process failures via deep learning. *Inventions*, 5(3), 2020. ISSN 2411-5134. doi: 10.3390/inventions5030025. URL https://www.mdpi.com/2411-5134/5/3/25.

[54] Yaser Banadaki, Nariman Razaviarab, Hadi Fekrmandi, and Safura Sharifi. Toward enabling a reliable quality monitoring system for additive manufacturing process using deep convolutional neural networks, 2020. URL https://arxiv.org/abs/2003.08749.

[55] Yuanbin Wang, Jiakang Huang, Yuan Wang, Sihang Feng, Tao Peng, Huayong Yang, and Jun Zou. A cnn-based adaptive surface monitoring system for fused deposition modeling. *IEEE/ASME Transactions on Mechatronics*, 25(5):2287–2296, 2020. doi: 10.1109/TMECH.2020.2996223.

[56] Yanzhou Fu, Austin R.J. Downey, Lang Yuan, and Hung-Tien Huang. Real-time structural validation for material extrusion additive manufacturing. *Additive Manufacturing*, 65:103409, 2023. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2023.103409. URL https://www.sciencedirect.com/science/article/pii/S2214860423000222.

[57] Mangolika Bhattacharya, Mihai Penica, Eoin O'Connell, and Martin Hayes. Ai-driven real-time failure detection in additive manufacturing. *Procedia Computer Science*, 232:3229–3238, 2024. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2024.02.138. URL https://www.sciencedirect.com/science/article/pii/S1877050924003168. 5th International Conference on Industry 4.0 and Smart Manufacturing (ISM 2023).

[58] Ikenna A. Okaro, Sarini Jayasinghe, Chris Sutcliffe, Kate Black, Paolo Paoletti, and Peter L. Green. Automatic fault detection for laser powder-bed fusion using semi-supervised machine learning. *Additive Manufacturing*, 27:42–53, 2019. ISSN 2214-8604. doi: https://doi.org/10.1016/j.addma.2019.01.006.

[59] Hermann Baumgartl, Josef Tomas, Ricardo Buettner, and Markus Merkel. A deep learning-based model for defect detection in laser-powder bed fusion using in-situ thermographic monitoring. *Progress in Additive Manufacturing*, 09 2020. doi: 10.1007/s40964-019-00108-3.

[60] Benedict O'Neill. Wevolver: Belt 3d printers explained: Technology, best machines, advantages. https://www.wevolver.com/article/belt-3d-printer, September 2022. Retrieved on December 26, 2024.

[61] Adam Fasnacht. Powerbelt 3d: How to orient & slice models for a conveyor belt 3d printer. https://powerbelt3d.com/how-to-orient-models-for-a-conveyor-belt-3d-printer/, 2024. Retrieved on December 26, 2024.

[62] Jackson O'Connell. All3dp: Belt 3d printer: All you need to know. https://all3dp.com/2/3d-printer-belt-all-you-need-to-know/, May 2022. Retrieved on December 26, 2024.

[63] ABB. Irb 6700. https://new.abb.com/products/robotics/robots/articulated-robots/irb-6700, 2025. Retrieved on January 7, 2025.

[64] RAW Idea. Tanaruz 3d-printed boats. https://tanaruz.boats/, 2024. Retrieved on March 19, 2024.

[65] Manos. Grabcad community, rib. https://grabcad.com/library/rib--3, May 2012. Retrieved on December 25, 2024.

[66] Luca Delbarba. Grabcad community, abb anthropomorphic robots. https://grabcad.com/library/abb-anthropomorphic-robots-1, February 2022. Retrieved on December 25, 2024.

Graphical Abstract

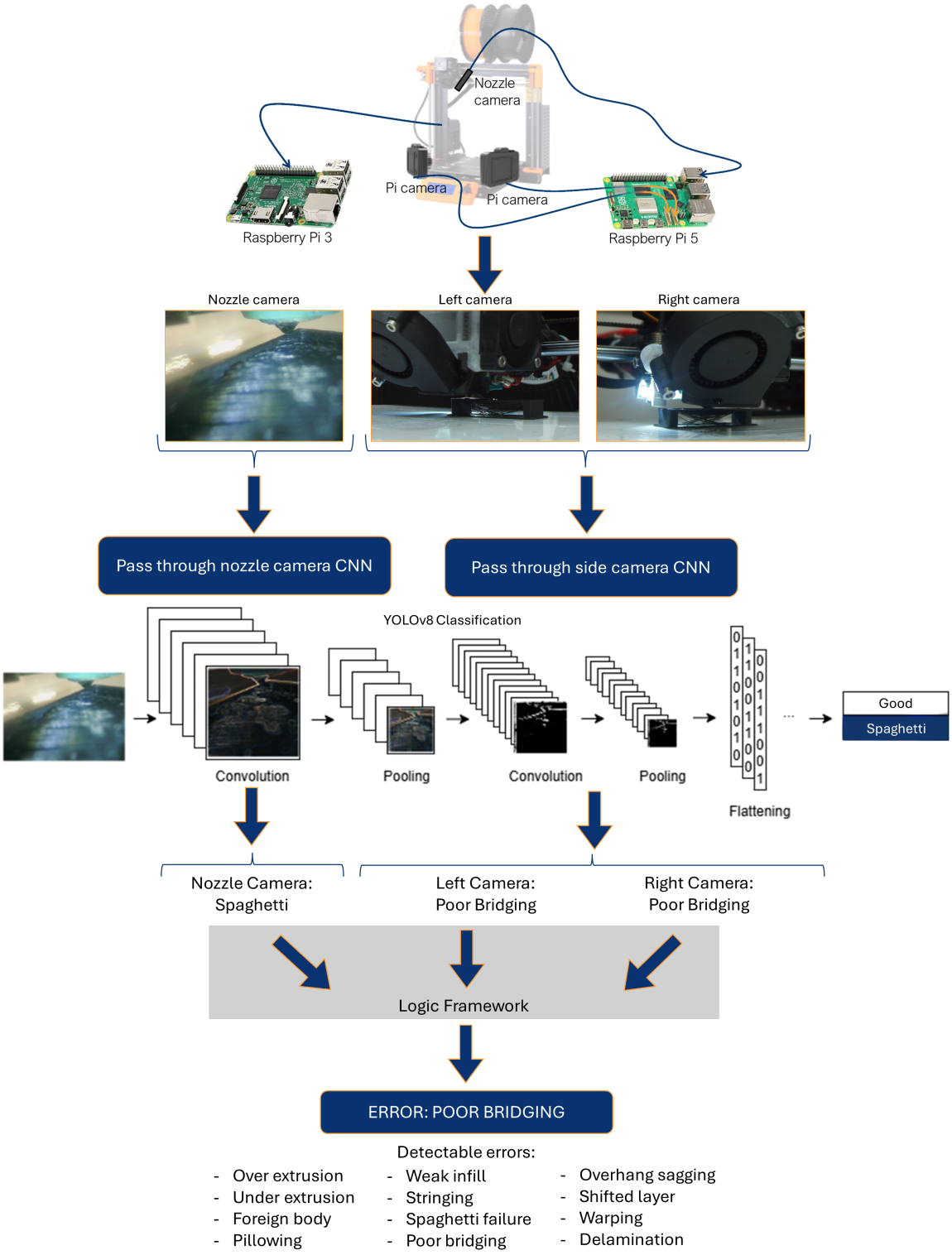


Figure 5.1: Graphical Abstract (subfigures from [32, 47, 48])

Cover Letter

Delft University of Technology
Mekelweg 5, 2628CN Delft, Netherlands
February 9, 2025

Dear Prof. A. T. Clare,

We are writing to submit the research paper titled 'Multi-Camera Fault Detection in Fused Deposition Modelling' for consideration to the journal 'Additive Manufacturing'. We confirm that this work is original and has not been published elsewhere, nor is it under consideration by any other journal.

This study presents a novel fault detection system for desktop FDM printers, capable of identifying 12 common error types. Our system integrates three cameras, one nozzle camera and two side cameras, and two convolutional neural network (CNN) models into a unified framework. The integration logic ensures real-time, in-situ error detection, enhancing the reliability and efficiency of the printing process.

The manuscript aligns strongly with the scope of Additive Manufacturing by addressing a critical challenge in additive manufacturing: error detection and mitigation. This research is an important initial step toward closed-loop control systems for FDM, laying the foundation for fully automated, real-time fault correction mechanisms. Additionally, it contributes to advancements in additive manufacturing processes by demonstrating a scalable detection system.

The key scientific step of this work is its multi-camera integration approach, which enables a more comprehensive detection system. While previous studies in the literature have typically relied on single-camera systems to detect a limited range of errors, our system identifies 12 distinct printing errors across multiple perspectives. This comprehensive coverage is achieved without disrupting the printing process, positioning this work as a significant step forward in real-time quality assurance for FDM printing.

We confirm that there are no conflicts of interest to disclose. Thank you for your consideration of this research paper. We look forward to your feedback and the opportunity to contribute to the continued advancement of the field.

Sincerely,
Aster Merel A. Tournoy, Shanthalakshmi Kilambi, Muhamad Amani, Dr. Jovana Jovanova, Dr. Kunal Masania

Article Highlights

- 3D printer fault detection system that detects 12 error types
- Multi-camera detection system including 1 nozzle camera and 2 bed cameras
- Two convolutional neural network models for error classification
- Integration of three camera outputs through logic framework

Figures

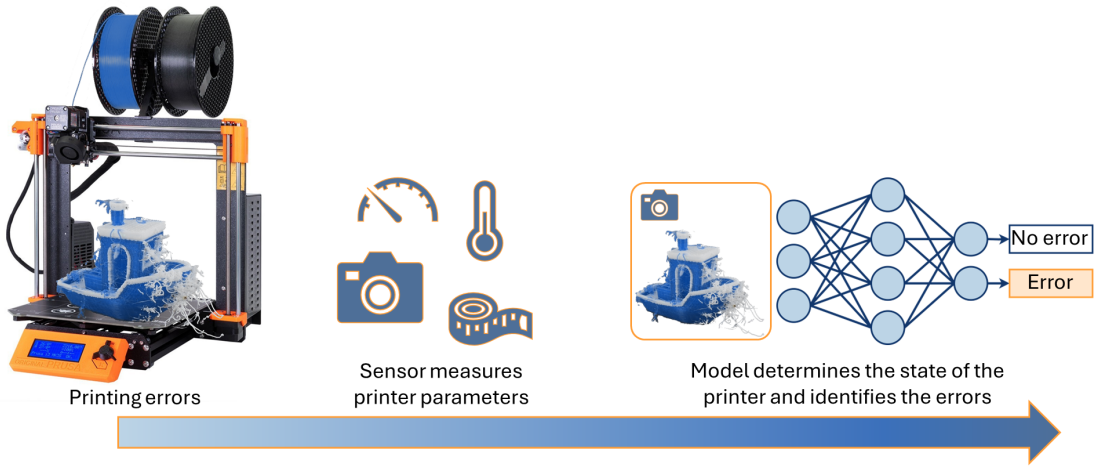


Figure 1.1: Fault detection system allows for monitoring the printer. Sensors are installed such that models can use the measurements to define and identify printing errors (adapted from [32, 33])

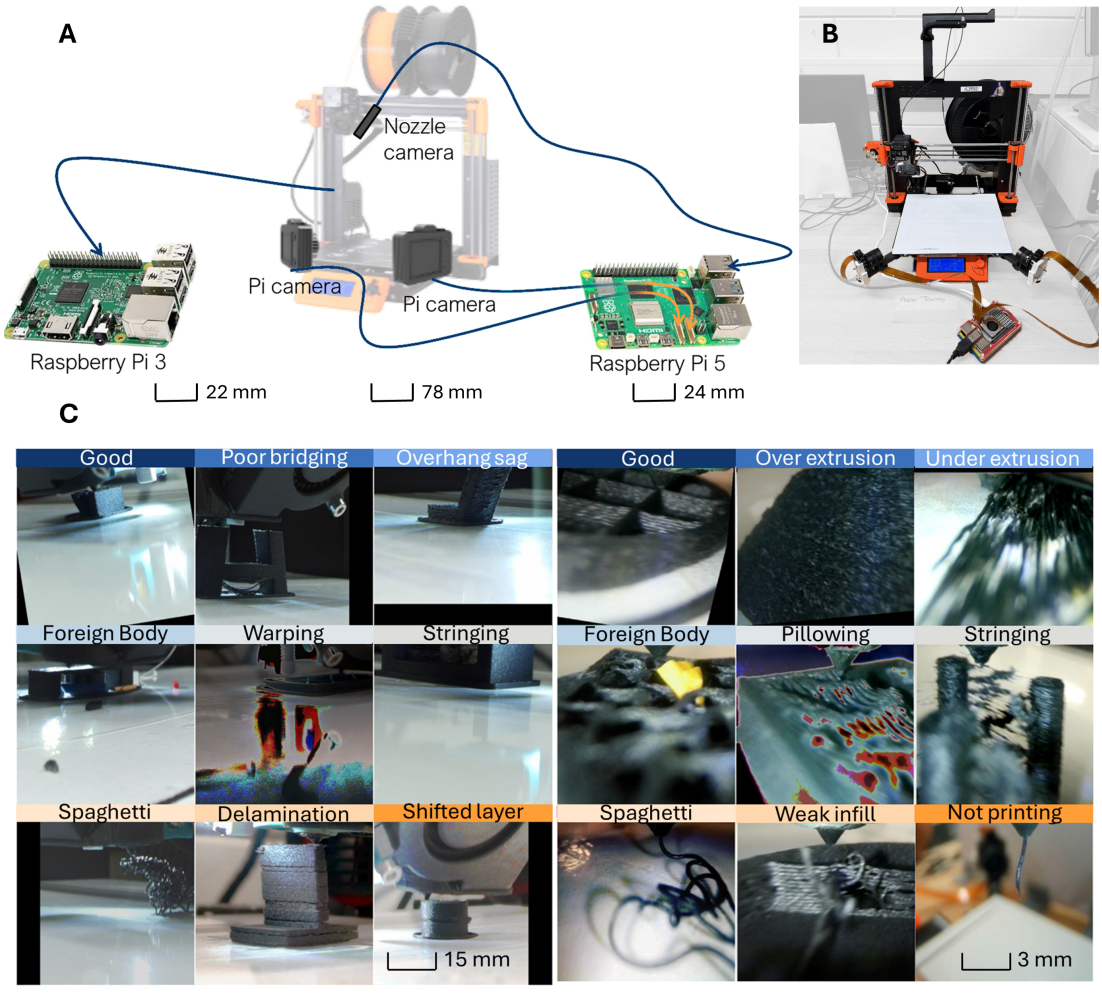


Figure 2.1: Data Collection. A: The graphical setup of data collection consisting of three cameras and two raspberry pis (adapted from [32, 47–49]). B: The physical setup. C: Data labelling and augmentation, left side shows the augmented images of the side camera, the right side shows the augmented images of the nozzle camera.

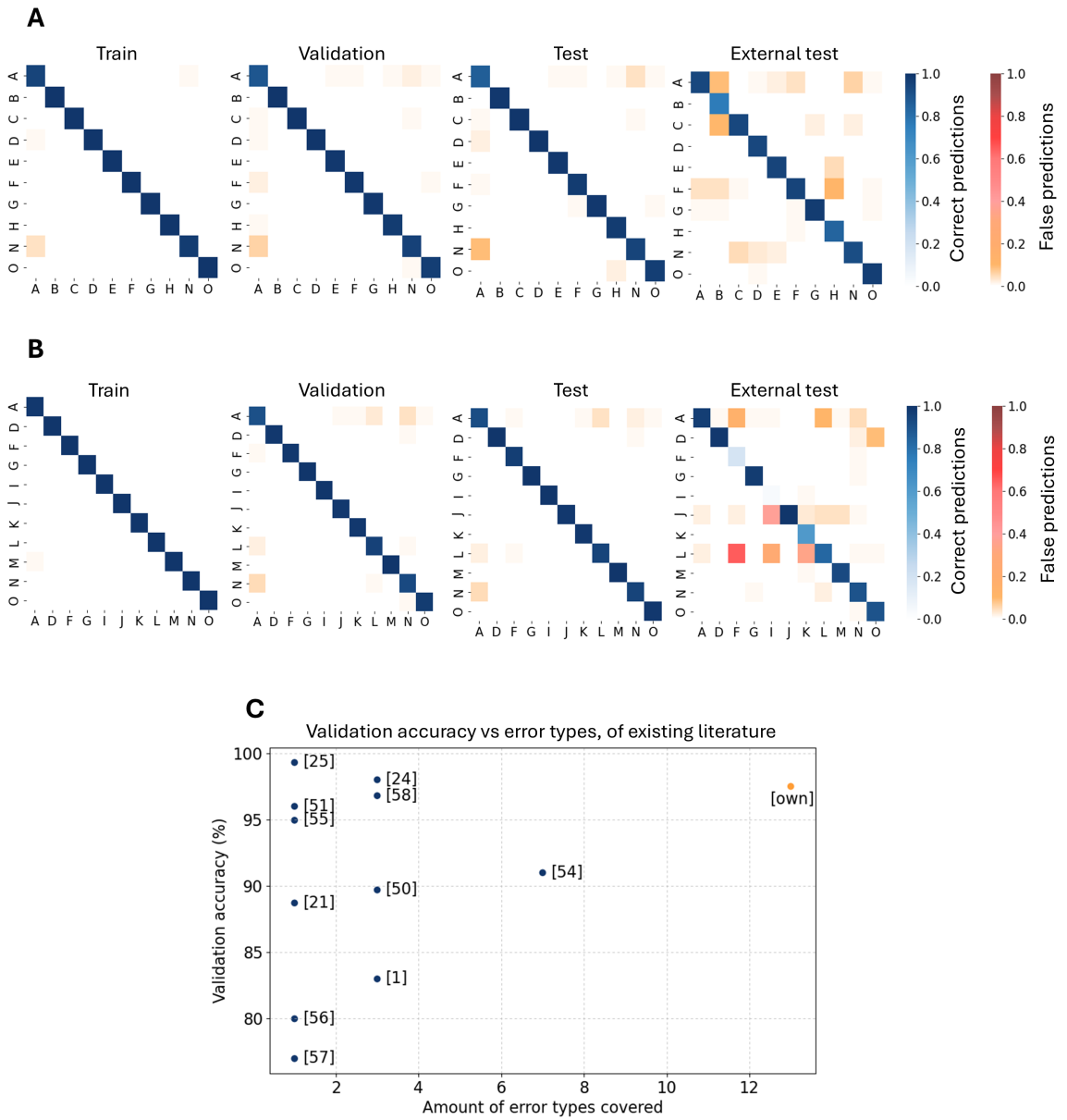


Figure 3.1: Results of the models. A: The confusion matrices of the nozzle camera model, for the training, validation, test and external test set, the individual classes are labelled according to the legend. B: The confusion matrices of the side camera model, for the training, validation, test and external test set. C: The validation accuracy of our model compared to existing literature, considering the amount of error types covered.

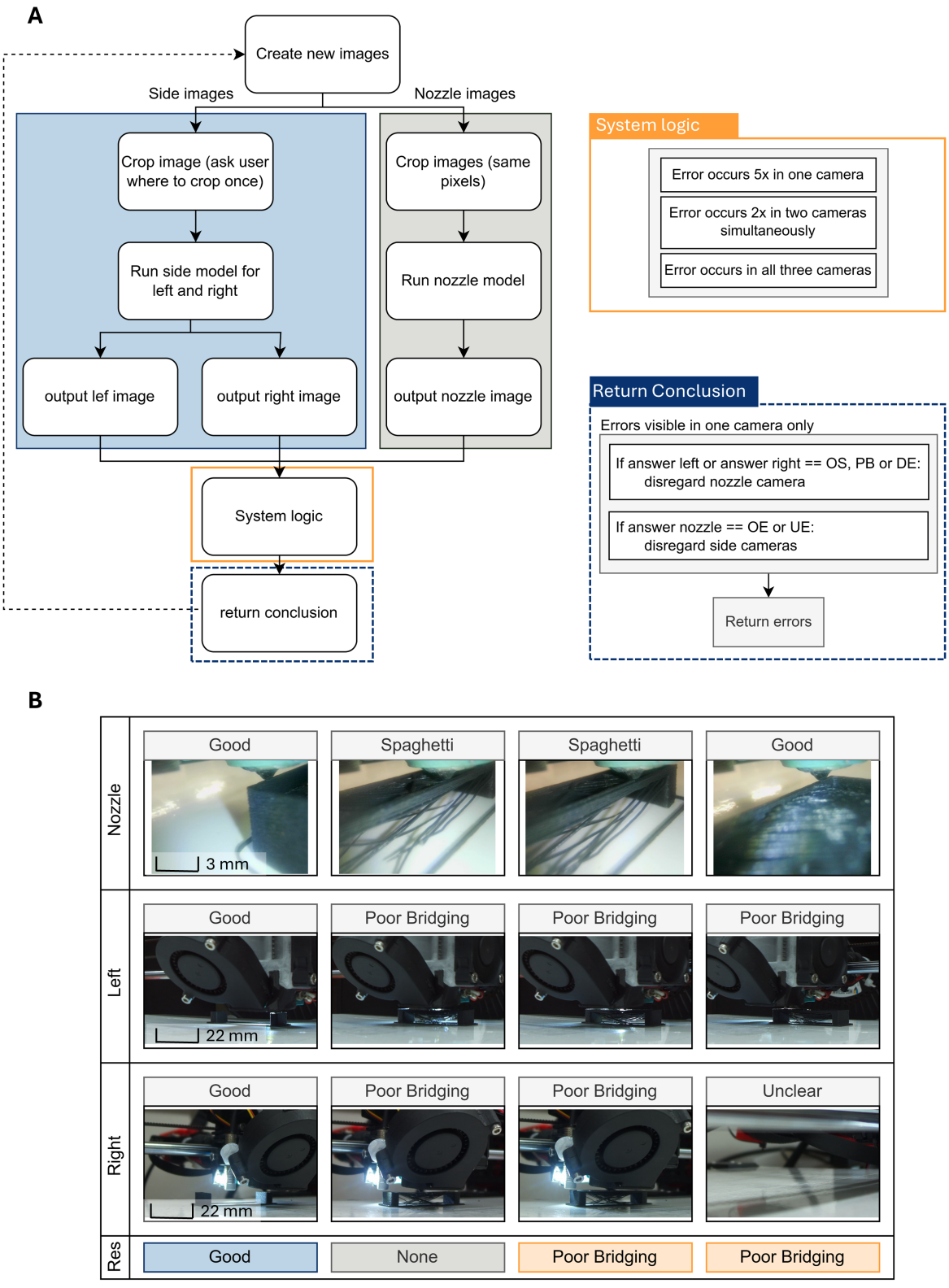


Figure 3.2: Fault detection system. A: Outline of the system, comprising of a cropping mechanism, applying the models, retrieving the outputs, applying a logic to those outputs to form a unified conclusion. B: Example case of the detection mechanism, where twice in a row two cameras detected the same error: poor bridging, hence that becomes the conclusion.

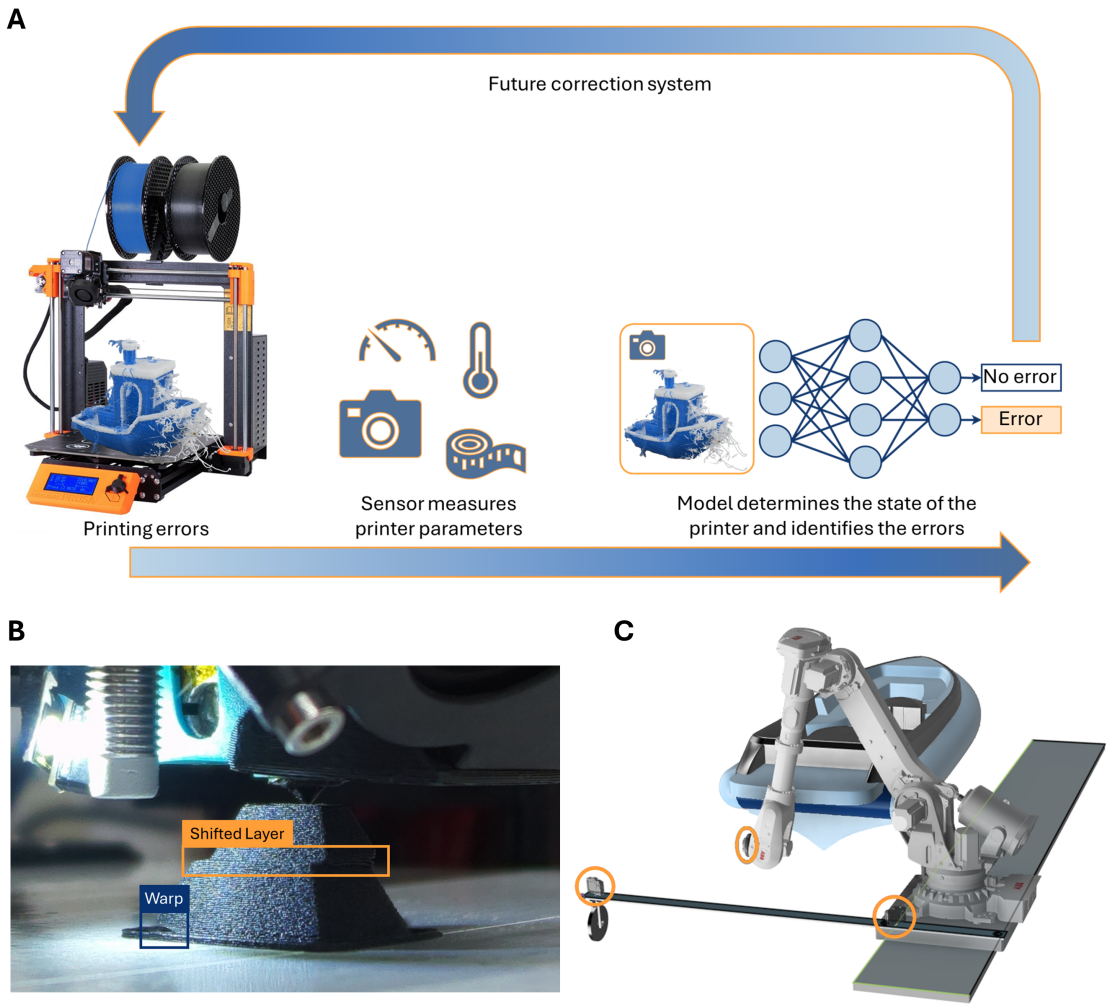


Figure 4.1: Future scope of the fault detection system. A: Making the process closed loop by integrating a correction mechanism. B: Object detection system can detect two errors simultaneously and locate errors, which allows for precise correction. C: The scope of the fault detection system can be extended to other FDM methods, such as large-scale FDM printing (adapted from [47, 65, 66])

Tables

Error group	Error	Detecting Camera
Extrusion Errors	Over Extrusion	Nozzle Camera
	Under Extrusion	Nozzle Camera
Detail Issues	Pillowing	Nozzle Camera
	Foreign Body in Print	Side Cameras
Structural Errors	Weak Infill	Nozzle Camera
	Poor Bridging	Side Cameras
Temperature Errors	Stringing	Nozzle and Side Cameras
	Overhang Sagging	Side Cameras
Alignment and Motion Errors	Shifted Layers	Side Cameras
	Spaghetti Failure	Nozzle and Side Cameras
Adhesion Errors	Warping	Side Cameras
	Delamination	Side Cameras

Table 2.1: Included error types and their detecting cameras

Metric	Training	Validation	Test	External Test
Accuracy (%)	99.11	97.68	97.09	93.18
Loss	0.07153	0.07464	0.07478	0.2770
F1 score (%)	99.11	97.66	97.10	91.09

Table 3.1: Performance of nozzle camera model: the accuracy, loss and F1 score of the training, validation, test and external test sets

Metric	Training	Validation	Test	External Test
Accuracy (%)	99.81	97.61	97.86	89.3
Loss	0.04255	0.1196	0.089906	0.03927
F1 score (%)	99.82	97.64	97.93	84.06

Table 3.2: Performance of side camera model: the accuracy, loss and F1 score of the training, validation, test and external test sets

Source	Printer	Sensor	Classes	Acc. (%)
[21]	FDM	Nozzle camera	nozzle height: high +, high, good, low	91.0
[23]	FDM	Nozzle camera	good, over, under extrusion	98.0
[15]	FDM	Nozzle camera	flow rate, speed, temp, Z-offset: low, good, high	96.6
[1]	FDM	Bed camera	normal, stringing, under extrusion	83
[24]	FDM	Bed camera	warping detection	88.72
[25]	FDM	Bed camera	warping detection	99.3
[51]	FDM	Bed camera	stringy, obstacle, unstick	89.7
[53]	FDM	Bed camera	acceptable, faulty	70.0
[54]	FDM	Bed camera	Print speed value, temperature value	94.0
[52]	FDM	Bed camera	spaghetti object detection	96.0
[55]	FDM	Rotating bed camera	normal, blob, void, thick line, crack, misalignment	91.0
[56]	FDM	Top-view camera	acceptable, faulty	84.95
[57]	VAT	Temperature, vibration, humidity	acceptable, faulty	80
[58]	L-PBF	Photodiode	acceptable, faulty	77
[59]	L-PBF	Thermographic camera	ok, metal splatters and delamination	96.8
own	FDM	Nozzle and 2 bed cameras	15 classes , containing 13 errors	97.7, 97.6

Table 3.3: Comparison of fault detection methods using NN in literature in additive manufacturing