NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

HUGO DO CARMO LOPES DOS SANTOS COSTA

BSc in Electrical and Computer Engineering Sciences

ANOMALY DETECTION IN FUSED DEPOSITION MODELING WITH DEEP LEARNING AND COMPUTER VISION

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
September, 2024

DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING

ANOMALY DETECTION IN FUSED DEPOSITION MODELING WITH DEEP LEARNING AND COMPUTER VISION

HUGO DO CARMO LOPES DOS SANTOS COSTA

BSc in Electrical and Computer Engineering Sciences

Adviser: Filipe de Carvalho Moutinho
Assistant Professor, NOVA University Lisbon

Co-adviser: Duarte Bragadesto
CEO, Dualtech

Examination Committee

Chair: Anikó Katalin Horváth da Costa
Assistant Professor, NOVA University Lisbon

Rapporteur: João Almeida das Rosas
Assistant Professor, NOVA University Lisbon

Member: Filipe de Carvalho Moutinho
Assistant Professor, NOVA University Lisbon

MASTER IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
September, 2024

Anomaly Detection in Fused Deposition Modeling with Deep Learning and Computer Vision

---

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.1.18) [42].

Acknowledgements

I would like to express my deepest gratitude to Professor Filipe Moutinho for the trust, guidance, and patience shown during the development of this dissertation. Your support has been invaluable.

To all the professors who have contributed to my academic journey, thank you all. There are too many to list, but each has played a crucial role in my personal and academic growth.

My heartfelt gratitude to the Dualtec team for their support and interest in my academic and professional development over the past three years. Special thanks to Lemos, Alexandre, Duarte, Luis, Claudio, and Andrian for your encouragement and assistance.

To my friends who have been present throughout this journey-Gui, Franco, Nuno Nogueira, Carlos, Miguel, Renato, Nena, Brito, Tomas Cavique, Pedro Cavique, Inóspita, Vaqueira, Caetano, Diogo, André, Nuno Martinho, Júlio, Pedro Ferreira- For different reasons and contexts, I am truly grateful for your camaraderie, support and companionship.

To my parents, Alexandra and Marco, and my brothers, Afonso, António, and Henrique- thank you for unwavering trust, understanding, and unconditional support over years.

Lastly, but certainly not least, to Maria João: "thank you" seems insufficient for all you have done. You support, listening ear, and unwavering presence throughout all the hardships means the world to me. Without you in my life this would not be possible, i definitely am a better man with you by my side.

Although this list may seem extensive, i am completely sure i have missed many names. To everyone i have mentioned and those i may have inadvertently omitted, thank you from the bottom of my heart. I hope we can all continue to grow together.

Abstract

The introduction of Industry 4.0 concepts has fueled the advancement of modern technologies such as additive manufacturing and artificial intelligence. Interchangeably referred to 3D printing, additive manufacturing has gained widespread popularity not only for facilitating rapid prototyping, essential for product development, but also for its ability to manufacture localized customized parts without the need for special tooling. An additional advantage stems from the low initial investment cost, which makes the technology accessible. This is the case with Fused Deposition Modeling 3D printers. Traditionally, the quality assessment and monitoring of active 3D printing jobs have required significant human intervention, which can be imprecise. Anomalies might be overlooked by the human eye, and in some cases, this approach can be destructive to the printed object, leading to a reprint job and increased material wastage. The present document discloses available state-of-the-art systems that confer automatic anomaly detection on ongoing 3D printing jobs through the use of deep learning networks and visual data. Furthermore, an enhancement to a state-of-the-art system which operates using this type of deep learning framework is proposed. It is theorized that the proposed modifications will improve the system's performance and yield favorable results. Additionally, to verify the developed system reproducibility and contribute to the existing dataset of the state-of-the-art system, a dataset containing material deposition images will be developed.

Keywords: Industry 4.0, Additive Manufacturing, Artificial Intelligence, Computer Vision, Deep Learning, 3D Printing, Quality Inspection

Resumo

A introdução dos conceitos da Indústria 4.0 impulsionou o rápido desenvolvimento de tecnologias como inteligência artificial e manufatura aditiva. A ampla adoção da manufatura aditiva, frequentemente referida como impressão 3D provem, não só, por facilitar prototipagem rápida, essencial para o desenvolvimento de novos produtos, mas também pela capacidade que demonstra para a fabricação local de peças personalizadas sem necessidade de ferramentas especiais. Uma outra vantagem da área tecnológica passa pelo seu baixo custo de investimento inicial, como no caso das máquinas 3D Fused Deposition Modeling. Tradicionalmente, a avaliação e controlo do decorrer de uma impressão 3D acarreta a intervenção de operadores humanos que visualmente inspecionam o seu progresso. Esta inspeção pode ser considerada defeituosa pois anomalias que não sejam detetadas pelo olho humano podem, em alguns casos, levar a destruição da peça impressa. O que requer uma nova impressão e gasto de material. O documento presente divulga sistemas de monitoramento automático da qualidade que utilizem inteligência artificial para analisar informação visual. É proposto uma melhoria para um sistema que se enquadra nas condições anteriores e é considerado ser o mais avançado, esta melhoria prevê que o sistema seja aperfeiçoado e produza resultados mais atrativos. Além desta modificação, é também desenvolvido um dataset que contém images de extrusão de material, com este dataset pretende-se testar a reprodutibilidade do sistema e contribuir para o dataset produzido no sistema selecionado.

Palavras-chave: Indústria 4.0, Inteligência Artificial, Manufatura aditiva, Visão computacional, Aprendizagem Profunda, Impressão 3D, Inspeção de Qualidade.

Contents

List of Figures

List of Tables

Acronyms

.cfg Configuration file format *(p. 55)*
.jpg Joint Photographic Experts Group image file format *(p. 53)*
2D Two dimensional *(p. 10)*
3D Three dimensional *(pp. 1–3, 9–11, 14, 16–18, 26, 27, 30, 31, 33, 35, 38, 42, 43, 54, 76, 82, 84)*

AI Artificial Intelligence *(pp. 1, 2, 5, 19, 21)*
AM Additive Manufacturing *(pp. 1–3, 5, 9, 10, 14, 27, 28, 33)*
ANN Artificial Neural Networks *(pp. 21, 23, 25)*

BoW Bag-of-Words *(pp. 29, 30)*

CAD Computer-Aided Design *(pp. 1, 2, 10)*
CAXTON Collaborative Autonomous Extrusion Network *(pp. 33, 44, 45, 47, 48, 50, 54, 63, 67–69, 71–73, 78, 83, 84)*
CIFAR Cannadian Institute For Advanced Research *(p. 19)*
CNN Convolutional Neural Networks *(pp. 2, 25, 29–32, 34)*
CPPS Cyber-Physical Production Systems *(pp. 6, 7)*
CPS Cyber-Physical Systems *(pp. 1, 6, 7)*
CSV Comma-separated values *(pp. 53, 54)*

DIY Do it yourself *(pp. 10, 42, 82)*
DL Deep Learning *(pp. 2, 3, 21, 24, 36, 53, 82)*
DLP Digital Light Processing *(p. 11)*

EBM Electron Beam Melting *(p. 13)*

FDM Fused Deposition Modeling *(pp. 1–3, 13–16, 26, 28, 30, 32, 35, 36, 38, 54, 84)*
FFF Fused Filament Fabrication *(p. 14)*

G-code	Geometric Code *(pp. 10, 14, 36, 55, 59, 61, 76)*
H2H	Human-to-Human *(p. 6)*
H2M	Human-to-Machine *(p. 6)*
I4.0	Industry 4.0 *(pp. 1, 5–9, 26)*
ILSVRC	ImageNet Large Scale Visual Recognition Challenge *(pp. 19, 48)*
IoT	Internet of Things *(pp. 1, 5)*
KNN	K-Nearest Neighbor *(p. 19)*
LCD	Liquid Crystal Display *(p. 11)*
LMD	Laser Metal Deposition *(p. 13)*
M2M	Machine-to-Machine *(p. 6)*
MJM	Multi-jet Modeling *(p. 11)*
MJPEG	Motion-JPEG *(p. 55)*
ML	Machine Learning *(pp. 2, 3, 19)*
MLP	Multi-layer Perceptrons *(p. 22)*
OpenCV	Open Computer Vision Library *(p. 58)*
PTFE	Polytetrafluoroethylene *(p. 16)*
RAMI4.0	Reference Architecture Model for Industrie 4.0 *(p. 8)*
RAN	Residual Attention Network *(pp. 44, 45, 47, 50)*
ReLU	Rectified Linear Unit Function *(pp. 22, 23, 48)*
RNN	Recurrent Neural Networks *(p. 24)*
RPF	Rapid Freeze Prototyping *(p. 11)*
SBC	Single Board Computer *(pp. 27, 31, 33, 43, 44, 64, 79–81)*
SELayer	Squeeze and Excitation Layer *(pp. 44, 48, 50–52, 69, 71, 72, 75, 83, 84)*
SiLU	Sigmoid Linear Unit Function *(p. 23)*
SLA	Stereolithography *(p. 11)*
SLM	Selective Laser Melting *(p. 12)*
SLS	Selective Laser Sintering *(pp. 12, 31)*
STL	stereolithography *(p. 10)*
SVM	Support Vector Machines *(pp. 19, 27, 28)*

Tanh Hyperbolic Tangent Function *(p. 23)*

UI User Interface *(p. 55)*

URL Uniform Resource Locator *(p. 56)*

UV Ultraviolet *(p. 11)*

1

Introduction

1.1 Background

Recent structural shifts in the industry paradigm have introduced new concepts and advanced the development of modern technologies that include Additive Manufacturing (AM), Artificial Intelligence (AI) and Internet of Things (IoT), which foster an environment equipped to meet the rapidly changing demands imposed on the sector. The Fourth Industrial Revolution, characterized as Industry 4.0 (I4.0), comes as the catalyst for the changes that are currently still being employed, introducing new strategies to approach the industry landscape, based on the seamless integration of cutting-edge technologies such as Cyber-Physical Systems (CPS) with already established systems that intend, but are not limited to, increase automation, efficiency, and productivity standards on a myriad of sectors as in Healthcare, Energy, Agriculture, and Economy. Production processes have also undergone disruptive structural changes throughout the value chain. From the stages of product conception and development to the end use of products, everything has been thoroughly modernized to comply with the guidelines of Industry 4.0.

An outcome of the I4.0 guidelines has been the further development of manufacturing techniques that enable fast production while at the same time conferring lower production costs as in the case of AM or interchangeably referred to as Three dimensional (3D) printing. The method pertains to the production of three-dimensional objects, designed with Computer-Aided Design (CAD), through the systematic deposition of materials in successive layers, which due to their method of fabrication allow for the production of objects with complex geometries otherwise unfeasible on traditional manufacturing processes such as Subtractive Manufacturing.

Among the many machines that comprise this manufacturing technique Fused Deposition Modeling (FDM) 3D printers are exceptionally attractive due to the minimal initial investment cost and lower material costs. Presently, the evolution of this category of machines is not solely driven by major corporations, but is additionally strengthened by the dynamic participation of open-source communities. These communities play a pivotal role in AM by developing innovative devices and architectures that improve the machines' printing quality and reliability, minimize printing time, and reduce material wastage from the printing process, therefore, redefining the fundamental standard expected of the machine printing capabilities.

3D printing quality is a critical aspect that heavily impacts the capability of a 3D printing machine, relates to both aesthetic and mechanical aspects of the final printed object. Aesthetic quality pertains to the visual attributes of the printed item, while mechanical quality involves the object's dimensional parameters and its tensile strength, in most cases the presence of anomalies that hinder quality often necessitate reprinting to fulfill the desired specifications. FDM printers are tremendously susceptible to errors that may arise from improper pre-printing calibration, mechanical problems or may be induced during the CAD model software processing stage, in this stage, numerous settings can be adjusted, each responsible for different aspects of the printing process. Some parameters however, are not constant and can be adjusted during runtime, which provides even further calibration capabilities to atone for small deviations from pre-calibration and increase printing quality.

Presently, quality assurance in 3D printing is maintained through pre-calibration techniques facilitated by computer models that evaluate numerous parameters. However, this process involves substantial human intervention, which can introduce errors if parameters are inaccurately configured for the specific printer-material combinations, leading to printing anomalies that can be destructive to the final product. Traditionally, print job monitoring has depended on thorough human visual inspection, which can be imprecise, as anomalies may go unnoticed by the human eye. Moreover, two significant challenges stem from reliance on human inspection: different errors may exhibit similar visual anomalies as some parameters are directly correlated, complicating the accurate identification of underlying causes, and for printing jobs that span hours, days, or even weeks, this reliance substantially diminishes the reliability of human inspection.

AI is a rapidly advancing technology that has demonstrated an exceptional capability to solve complex tasks, which would otherwise be time-consuming and or even inconceivable. The emergence of sophisticated Deep Learning (DL) models have enabled the processing and analysis of visual data. Convolutional Neural Networks (CNN) models are widely utilized for tasks involving image recognition and anomaly detection. Their ability to capture and evaluate intricate relationships on visual data is particularly relevant to quality assurance on FDM 3D printers, where anomalies that hinder printed items may be caused by a combination of factors that are not easily recognizable by human operators. Machine Learning (ML) models are already being integrated into AM machines, yielding promising outcomes. Researchers have devised novel architectures that leverage cameras not only to monitor print jobs but also to automatically capture and evaluate images of the manufacturing process for the presence of anomalies. The integration of artificial intelligence models in conjuncture with FDM 3D printers proves to be a valuable tool to create a standard procedure for automatic error detection. Therefore disrupting the technological sector even further by providing 3D printers with more resilient systems capable of generating reports during the printing process that can be used to determine error causes and even taking action to stop the printing process.

1.2 Objective

As mentioned on the previous section, additive manufacturing technologies, particularly FDM 3D printers, lack a standardized method for assessing the quality of printed structures. Traditional methods, such as relying on human inspection, are inherently flawed. This becomes particularly evident when the production of a 3D structure takes days or even weeks to complete. To ensure the quality of the printed object, human operators would be required to consistently monitor the printing process. Additionally, the diagnosis of anomalies that affect the quality of printed objects may be incorrect, as an anomaly can be caused by different issues, each requiring an appropriate solution.

The complications previously outlined necessitate measures to enhance the reliability of FDM 3D. Among the extensive literature on systems that provide automatic quality assessment of 3D printed items, one particular solution stands out with promising results. This solution leverages DL networks to continuously evaluate the presence of anomalies in filament deposition images gathered from active printing jobs.

A particular system, utilizing this architecture to evaluate the quality of active printing jobs has shown state-of-the-art results when investigating systems that operate with leverage such frameworks. The outcomes are attributed not only to the designed model but also to the consideration of the unique interdependencies among various printing parameters to identify anomalies which is considered to be a crucial feature as it addresses one of the previously discussed issues.

Taking into account this cutting-edge system, the current study aims to make contributions in the following ways:

1. **State-of-Art**: Disclose pertinent information that related to AM technologies, followed by an in-depth view of FDM 3D printers, ending with anomaly detection systems that leverage ML models.

2. **Creation of a Dataset**: Develop a dataset consisting of material deposition images that considers the interdependencies between printing parameters and anomalies. The main goal behind this dataset is to expand the dataset produced by the development of the system that is considered to be state-of-the-art. An additional objective is to determine the reproducibility on different, more common types of FDM printers.

3. **Anomaly Detection System**: Propose a modification to the previously highlighted systems DL model, with the goal of enhancing the model's performance.

1.3 Document Structure

The document is structured as follows to present the work carried out:

Chapter 1 - Introductory chapter that discloses the dissertation background, objectives and the dissertation structure;

Chapter 2 - Discloses pertinent information with supporting concepts that are the cornerstone of which the development of this dissertation is based upon. Additional research is divulged which is directly related to the proposed work;

Chapter 3 - An in-depth view of the developed system, with step-by-step explanations;

Chapter 4 - Results acquired from the development of the proposed work are exposed with considerations;

Chapter 5 - Final chapter concludes the current document with notes on the work developed, suggestions for future work, and a list of the main contributors.

2

Literature Review

The present chapter intents to disclose relevant information that is directly related to the development of the current research work. Revealed information is organized in a structured manner, segmenting pertinent sections that expose concepts and technologies which support the objective outlined in Section 1.2.

2.1 Industry 4.0

The evolution of the industrial paradigm has been marked by different revolutions that shaped humankind and with each revolution, changes across areas such has technology, economy and social structures shaped reality to be what we presently know. Around the nineteenth century, humankind experienced the First Industry Revolution, fueled by water and steam power, marked the transition from manual production methods to mechanized production; At the beginning of the twentieth century, society was impacted by the addition of mass production and widespread use of electricity dramatically changing the economic and social landscape which was denominated The Second Revolution; The Third Revolution, circa 1970, integrated computation and automation into the industry increasing production and efficiency [84].

We live in a rapidly changing technological state of advancements where the number of years passed between each industrial revolution is continuously reducing. Today, we are in the midst of the Fourth Revolution that has been modernly characterized as *Industry 4.0 Industry 4.0 (I4.0)*, term imported from an high-tech project by the German government originated in 2011 and adapted from the German term *Industrie 4.0* [84, 43, 61]. The Fourth Industrial Revolution, enabled by modern advanced technology, introduces fundamental shifts in the way we live, work and relate to one another. Fueled by breakthroughs in various modern advanced technologies that enabled this transition, the development of Artificial Intelligence Artificial Intelligence (AI), Big Data, Internet of Things Internet of Things (IoT), Digital Twin, Cloud Technologies, Cybersecurity, Autonomous Robots and Additive Manufacturing Additive Manufacturing (AM) are among the primary contributors and decisive forces to drive I4.0 [84].

Within the vast literature about Industry 4.0, Shafiq et al. [69] defines I4.0 as the combination of ”intelligent machines, systems production, and processes to form a sophisticated network” (p.1147). Xu et al. [84] additionally states that ”Industry 4.0 refers to the intelligent networking of machines and processes for the industry based on CPS - a technology that achieves intelligent control using embedded networked systems”(p. 531). They observe that Industry 4.0 phenomenon involves the deployment of devices with perpetual communication capabilities that achieve an higher degree of seamless integration between physical systems and digital technologies. This integration promotes the creation of an environment that nurtures sophisticated automation, adaptation, efficiency, and facilitates real-time data exchange through a network of intelligent machines, across the entire value chain [73, 16, 61].

To achieve this, *Cyber-Physical Systems (CPS)* are considered to be one of the backbones of I4.0 [84]. These systems are geared towards promoting connectivity and employing the merger of the physical and digital realms with computational abilities. This aspect enables the collection and analysis of data from physical monitors, such as sensors, for an in depth specialized localized knowledge of a system. Leveraging the data collection while promoting a fusion with computational power, actuators and communication capabilities produces a unique array of characteristics that confers a machine with a high degree of autonomy and power to react to physical situations [69, 61]. The application of these systems on the manufacturing realm introduce *Cyber-Physical Production Systems (CPPS)*, which is a specialized application of the aforementioned CPS devices on manufacturing plants. The objective of this integration was to create a degree of modularity on the manufacturing processes increasing optimization with the main goal of improving production and reducing downtime of the manufacturing plant [69]. The applicability of CPS systems is not limited to the manufacturing landscape. The employment of such devices can also improve a wide range of areas including healthcare, agriculture, transportation, energy, and others[16].

The I4.0 concept has a much broader range, and cannot be limited only by the introduction of modern technologies, but instead as a strategy and a group of design principles that create a structural change which spans across the entire value chain [61, 69]. Vogel-Heuser and Hess [73], Shafiq et al. [69], Lu [43] and Bundesverband Informationswirtschaft et al. [8] enumerate the design principles that should be considered as key points for an implementation strategy of Industry 4.0 I4.0:

- **Interoperability**: Represents the ability of the participants to seamless exchange information, which can be between *Human-to-Human (H2H)*,Machine-to-Machine (M2M) or Human-to-Machine (H2M). This point is crucial for implementing a responsive and efficient environment.

- **Decentralization**: The ability of CPS & CPPS and humans to make informed decisions on their own without the approval of a central authority. This leads to reducing the downtime by having real-time adaptations.

- **Virtualization**: Creation of *Digital-twins*, virtual copies of real-life devices that enables behavioral analysis of such devices or humans. This can improve the performance due to the fact that many different scenarios can be tested and based on the results of the *Digital-twins* changes can be made to the process leading to a general improvement.

- **Real-time Capabilities**: CPS & CPPS devices enable real-time data acquisition and analysis, which permits real-time information processing and analysis, reducing downtime and providing possible optimizations.

- **Modularity**: The ability to adapt by increasing or decreasing individual modules to meet evolving demands and requirements without affecting the rest of the manufacturing unit maintaining production.

- **Service Orientation**: Describes the ability for improved online-services that can offer integrated products that can bring bigger value for the customer.

As part of the I4.0 implementation strategy, the aforementioned design principles should act in three critical topics [8, 16, 69]:

- **Horizontal integration**: Relates to the use of systems across the entire value network that enhance communication between agents. In this topic it is expected that the various members that compose the value chain, such as manufacturers, logistics companies, engineering, marketing and so forth, create an environment of network with collaboration and information sharing. This integration offers an end-to-end solution that makes interactions flexible to the different needs along the entire value chain. Additionally, Horizontal Integration is expected to create new business opportunities with the inclusion of small and medium-sized into the network leading to the creation of new business models.

- **End-to-end integration**: Concerns to the engineering and development over the entire product life-cycle. The idea behind this sets on all agents to be involved in a products life-cycle until it reaches its life end.

- **Vertical integration**: Refers to the integration of intelligent systems such as actuators, sensors, calibration systems, and control points with processing units that are interconnected across the different hierarchical levels that compose manufacturing systems.

The need to solidify the understanding of all these different aspects that together compose I4.0, a simple yet structured reference guide is necessary. *Plattform Industrie 4.0* group; composed by three German associations BITKOM, VDMA and ZVEI; began to reunite all findings with the help of several institutions with the purpose of creating a standard. The outcome of this collaboration resulted in the development of a uniform reference architecture model called *Reference Architecture Model for Industrie 4.0 (RAMI4.0)*, which is portrayed in Figure 2.1, containing all fundamental aspects of I4.0 with the life-cycle and value stream structured in an hierarchical approach. Thus RAMI4.0 can be considered as the reference model for the implementation and description of the different production components that compose I4.0.[8, 16].

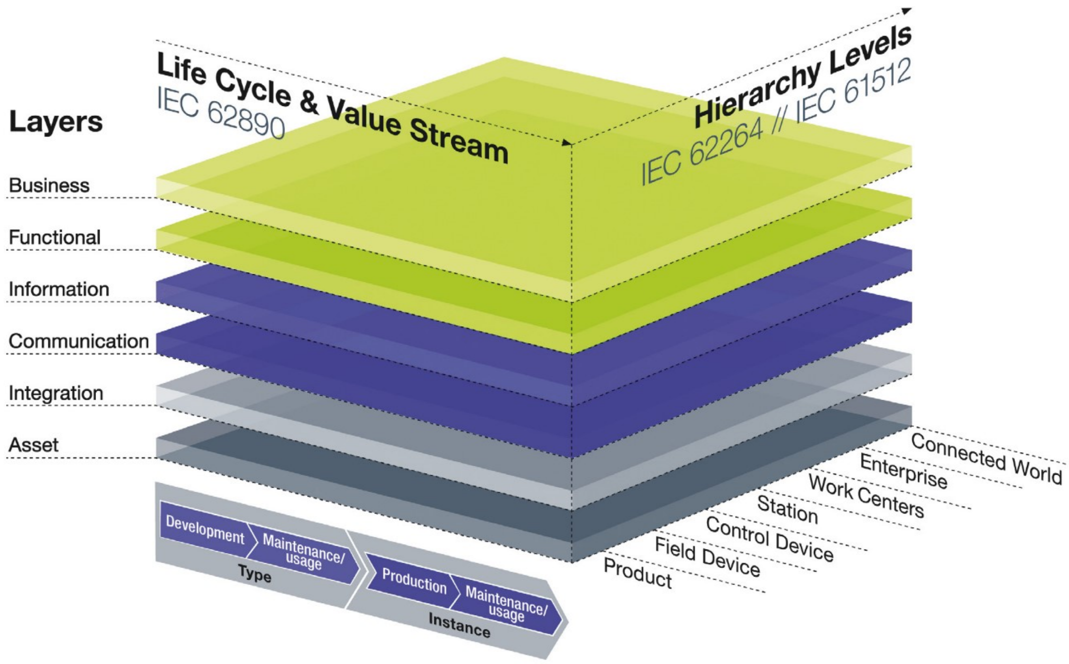


Figure 2.1: The Reference Architecture Model for Industrie 4.0 [8].

RAMI4.0, Figure 2.1, depicts a three-dimensional representation of I4.0 systems that can be described as follows [84, 73, 69, 16, 64, 8]:

- ”**Life Cycle & Value Stream**”: Placed on the horizontal axis, represents the relationships between life cycle with their correspondent value streams, which include factories, products, costumers, machines and everything that is product related. Depicts the notions of perpetual data acquisition and sharing throughout the life cycle of a component. This creates room for improvement by gathering information about product performance and sales, which can lead to subsequent updates of the product. The main purpose of this representation is to describe the importance of data sharing across all components that represent the life cycle and value stream of a component, with a clear end goal of capturing potential for flexible adaptation and improvements.

- ”**Layers**”: Represented on the vertical axis, combines various perspectives with a degree of abstraction. Emphasizes the need of digitization of real-world assets into a virtual representation, with the goal of creating a controlled environment, where all data that is available is used to control and create information to be available on all processes. All individual components that form the Layers axis must be adapted to be compatible with the rest of the participants. Portrays the communication, as well as assets on the bottom level, business processes highest abstraction.

- "**Hierarchy Levels**": The third axis, gives a representation that aggregates all functional components present in an industrial environment. The flow throughout the hierarchical levels lead to the top end "Connect World", indicating that the intent of this axis is not only depict factory level components, but take it one step further and introduce the notion of connected factories and collaboration between all agents, consequently going beyond and above enterprise level representations and reaching the end of the chain. Ultimately reaching the point were everything is connected.

2.2 Additive Manufacturing

The I4.0 standard establishes a comprehensive set of guidelines and trends aimed at optimizing production processes and supply chain management, as previously discussed on Section 2.1. The concept is based on the seamless interconnection of various production processes that increase the overall efficiency. This interconnection facilitates reduction of material wastage and mass customization, which enables manufacturers to swiftly respond and adapt to rapid market changes and demands. Driven by the aforementioned guidelines, manufacturing processes are in a transformative state. From conventional manufacturing processes which include subtractive, joining, dividing and transformative manufacturing, a distinct type of manufacturing process technique stands out [2].

This technique referred to as Additive Manufacturing (AM), interchangeably referred to as 3D printing, differs from traditional manufacturing methods as the construction of three-dimensional shapes Three dimensional (3D) is achieved through consecutive layering of material, fundamentally opposed to subtractive manufacturing where material is removed to achieve the final object.

The method adheres to some definitions set by the I4.0 standard, key aspects of the technology include: fast prototyping, ability to manufacture a part in a relative short time span, making it crucial for product testing purposes; cost-efficiency, this manufacturing technique enables small-scale and low volume production satisfying the need for any amount of quantity. Additionally, with this technique there is no need for specialized tooling such as molds or jigs, 3D printing does not require such tools. Therefore objects can be obtained with reduced labor costs and material wastage. Furthermore, depending of AM technology used, initial equipment investment is relatively low and the materials can also be found to be low-cost [85, 24]. Production of complex geometries, due to the fabrication method, are easily achieved with no specialized tools and expensive equipment [24, 2].

Advancements on AM landscape were facilitated by advancements on Computer-Aided Design (CAD) software, which enables designers to effortlessly create complex three-dimensional shapes on computers as depicted on Figure 2.2. Contemporary CAD software facilitates the exchange of information throughout development teams by leveraging cloud technologies. An example of this is found within Open-Source 3D printing community, where the creation, sharing and collaborative development of 3D models are actively promoted. There are several platforms designed with this purpose in mind; *Thingiverse* and *Printables* are two prominent examples of online repositories that offer three-dimensional models available for download. Moreover, the exchange of information has significantly influenced the advancement of 3D printer solutions and components within the open-source community. *Voron Design* project is one example, originated in 2015, with the goal of developing Open-Source Do it yourself (DIY) 3D printer [74].

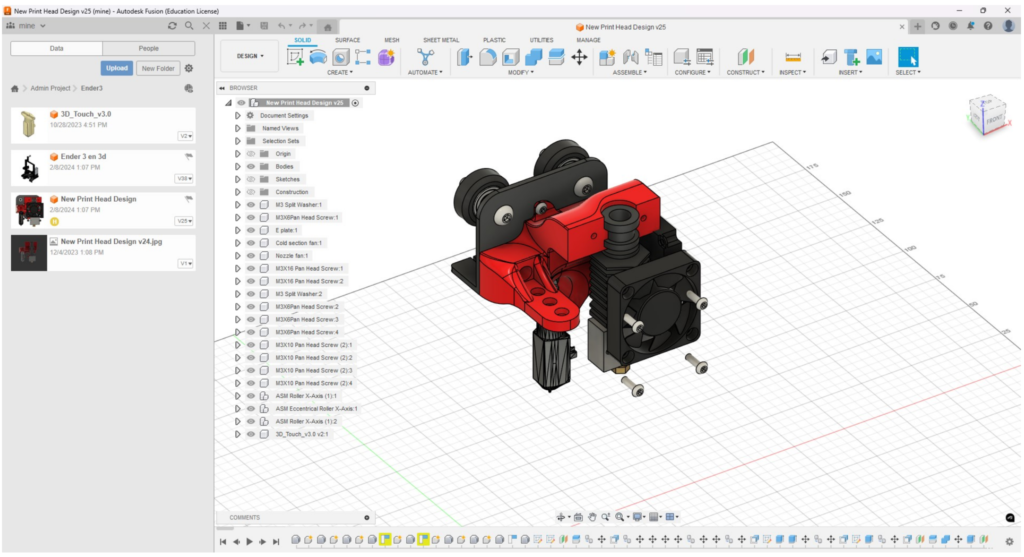


Figure 2.2: Example of a Computer-Aided Design (CAD) model designed in Autodesk Fusion 360.

Following the creation of a model in CAD, the information is then translated into a file format that converts the three-dimensional model into geometric data. The majority of downloadable open-source 3D models are available in the *stereolithography (STL)* file format, which emerged at 3D Systems by Chuck Hull. The format translates CAD models into a data representation that consists of triangular surfaces and vertices, which cover the entire CAD model. As of the moment, CAD software solutions have widespread coverage for this format [2, 81].

An additional software is required to convert raw 3D geometric data to instructions for the 3D printer to comprehend. *Slicing* software is tasked for this conversion by slicing the 3D model data into Two dimensional (2D) layers which convey movement instructions for the printer to follow, stacking these layers produce the final 3D structure. The conversion process produces a Geometric Code (G-code) file, a common programming language that conveys the instructions for a machine controller to interpret [1]. During this conversion stage, an extensive set of parameters can be configured to control the generated instructions, ensuring they are tailored to the specific mechanical configuration of the 3D printer is use. Figure 2.3 illustrates a simple workflow for the 3D printing process, highlighting the four fundamental key steps previously discussed.

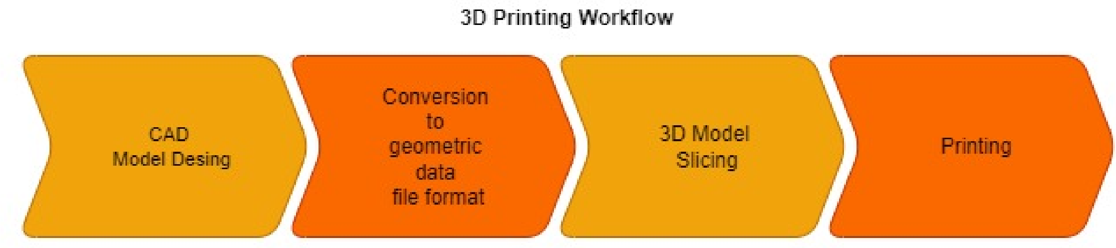
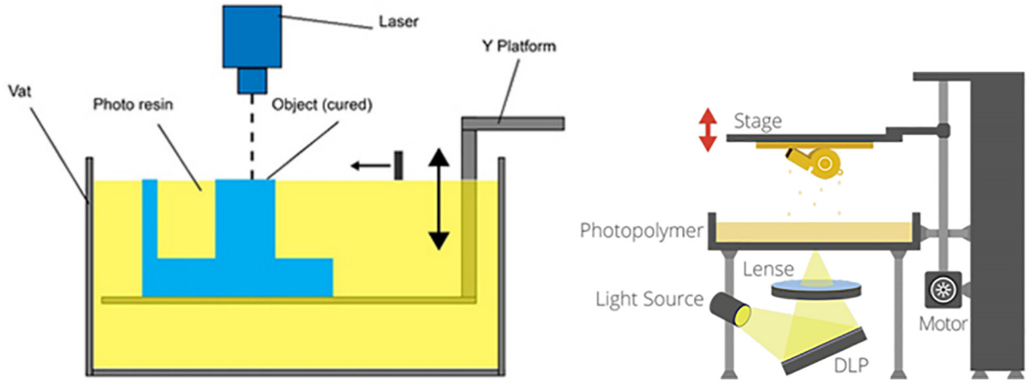


Figure 2.3: 3D printing workflow.

Additive Manufacturing processes have the capability to leverage a wide variety of materials in different physical states to construct three-dimensional objects. Consequently, a more effective method to categorize the extensive variety of printer classes is by classifying them based on the type of material with which each printer operates, as materials directly influence the fabrication methodology that each machine employs. Three main categories can be deduced from the available material types: liquid, solid and powder [2, 57].

Printers which handle liquid materials use a source of Ultraviolet (UV) light to cure and harden the material are are part of a class of *Vat polymerization techniques*. These materials are a type of photopolymer resin that solidifies upon exposure to UV light. ***Stereolithography (SLA)*** printers are part of this category, in which liquid material is solidified in a layer-by-layer manner. Figure 2.4(a) depicts this machine, they include a bed plate connected to a linear carriage that dips the plate into a reservoir filled with photopolymer resin. Each dipping maneuver cures the resin using a laser positioned at the bottom of the container, the laser then emits images layer-by-layer which cures the resin with the intended dimensions. A post-printing process is required to harden the printed part, this last stage entails exposing the printed part to additional UV light [2]. ***Multi-jet Modeling (MJM)***, Figure 2.6, is another example of this type of printers, by leveraging multiple nozzles which spray photopolymer liquid to a moving platform. A flash of UV light cures the liquid as it's being sprayed. Apart from these two examples, other machines follow the same principle of operation including *Rapid Freeze Prototyping (RPF)* and *Digital Light Processing (DLP)*, illustrated in Figure 2.4(b), which only differs from SLA by the type of UV light source, in this case the light source comes from a digital projector. Furthermore, a Liquid Crystal Display (LCD) can be used to as a source of light [24, 83].



(a) SLA printing process.
(b) DLP printing process.

Figure 2.4: Stereolithography (SLA) & Digital Light Processing (DLP) processes (gathered from [2, 46]).

Powder Bed Fusion processes utilize powder materials that extend beyond just polymers (plastics) to include metals, ceramics and composites, which can also be applied to fabricate three-dimensional objects. ***Selective Laser Sintering (SLS)***, illustrated in Figure 2.5, machines are trending in this category, demonstrating an exceptional capacity to manufacture strengthened high quality parts. The fundamental principle of operation is based on exploiting high power lasers that selectively melt powder material to create layers of material on the bed of the printer. After a layer is created, the bed platform moves down in preparation for the next layer, powder is spread on top of the piece and the processes repeats until the desired structure is finished. The technique involves a post-print cleaning process where excess powder is removed, most of the excess of powder cannot be reused, around 70-80% being one of the main disadvantages of the technology.

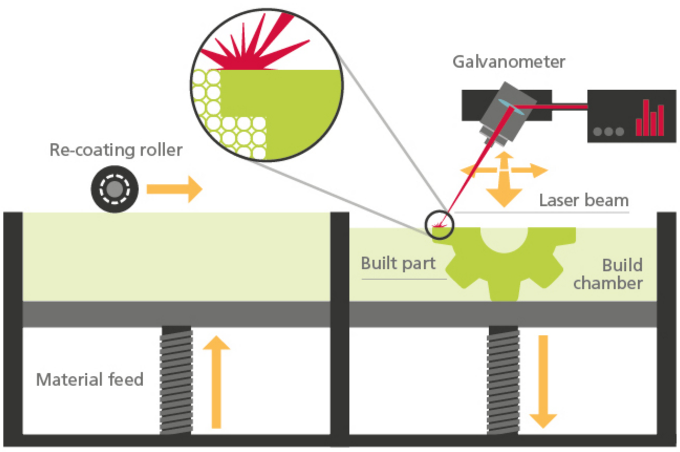


Figure 2.5: Selective Laser Sintering (SLS) printing process (gathered from [68]).

Another interesting method is **Selective Laser Melting (SLM)**, similarly to SLS, the machine takes advantage of high power lasers to melt metal powders in order to make high density parts. A pivotal challenge with this type of machine arise from the presence of residual stress making printed items susceptible to cracks due to deformities such as porosity within the solidified layers [2, 35, 57]. Additional machines that leverage powder materials include Inkjet (3DP), Laser Metal Deposition (LMD), Electron Beam Melting (EBM).

Binder Jetting printers are a specialized form of the powder bed fusion machines. This type of printer uses an hybrid material approach by taking advantage of powder and liquid materials to create three-dimensional structures, the difference lies within layer fusion, a liquid agent is utilized to bind layers of powdered material [79, 26]. .

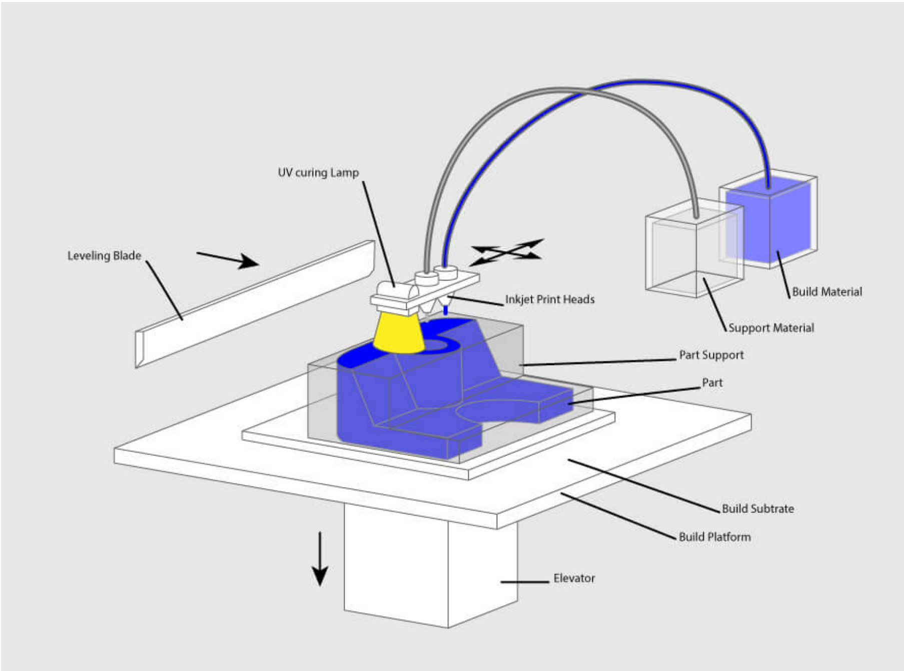


Figure 2.6: Multi-jet Modeling printer process (gathered from [79]).

Machines that operate using solid materials fall under the category of *Material Extrusion* technique . Among the multiple printers that use solid polymer materials, the most significant and pertinent to the current work is *Fused Deposition Modeling (FDM)* printers, due to their importance, the technology will be elaborated in greater detail on the following Section 2.3.

2.3 Fused Deposition Modeling

Fused Deposition Modeling (FDM) machines, also referred to as Fused Filament Fabrication (FFF), pertain to a specific machine class in AM that is arguably the most popular on a consumer level given their low initial investment cost.

Materials utilized in FDM machines are solid single thread derived from raw synthetic polymers (plastics), given their visual aspect, these materials are commonly referred to as *filaments*, and have across their length a controlled diameter of $1.75mm$ or $2.85mm$, correct dimensions are crucial as variations in diameter will produce visual deformities on printer parts. Filaments can be sub-categorized by the type of used polymer, different chemical constitutions exhibit distinct requirements settings namely temperature, print speed, ambient temperature and additional properties when being utilized for printing. Furthermore, filaments can be imbued with an extensive list of other material types such as wood, carbon and marble which produce materials called composites. Although filaments are the most commercially popular form of source material used on FDM printers, there are also alternatives that require different hardware on material extrusion machines, particularly raw *pellets* or pastes.

FDM printers produce 3D structures based on deposition of material in a layer-by-layer fashion, depicted on Figure 2.7. To achieve this end, the printer adheres to a coordinate system and deposits material to create one layer, progressive layer stacking produce a final three dimensional shape. The mechanical constitution of this type of printer lies on five main components: controller board, toolhead, extruder, bed plate, and motion system.

Figure 2.7: FDM printing process (gathered from [25]).

The toolhead, Figure 2.8, is a pivotal component of this mechanical architecture responsible for handling the liquefaction of material, preparing it for extrusion, often referred to as *Hotend*. The component is incorporated onto the motion system. Instructions provided by G-code file relate to the positions where the toolhead should move to produce the desired 3D structure. Four pivotal interlocking parts compose the toolhead, by the natural progression of the input material they are: the *heatsink* , *heatbreak*, *heating block* and finally the nozzle.

- **Heatsink**: Ensures a zone where the input filament remains with lower temperature by spreading and dissipating heat, this zone ensure that filament stays solid enough to control extrusion movements;

- **Heatbreak**: Couples the heatsink and heat block, a metal component that ensures a separation between lower temperature zone on the heatsink and the high temperature zone created by the heat block. This separation is crucial to ensure that no extrusion problems arise [20];

- **Heat block**: Responsible for the liquefaction of material, coupled together with a thermistor and a sensor that enables temperature control;

- **Nozzle**: The final component, where the liquefied material is expelled for deposition.

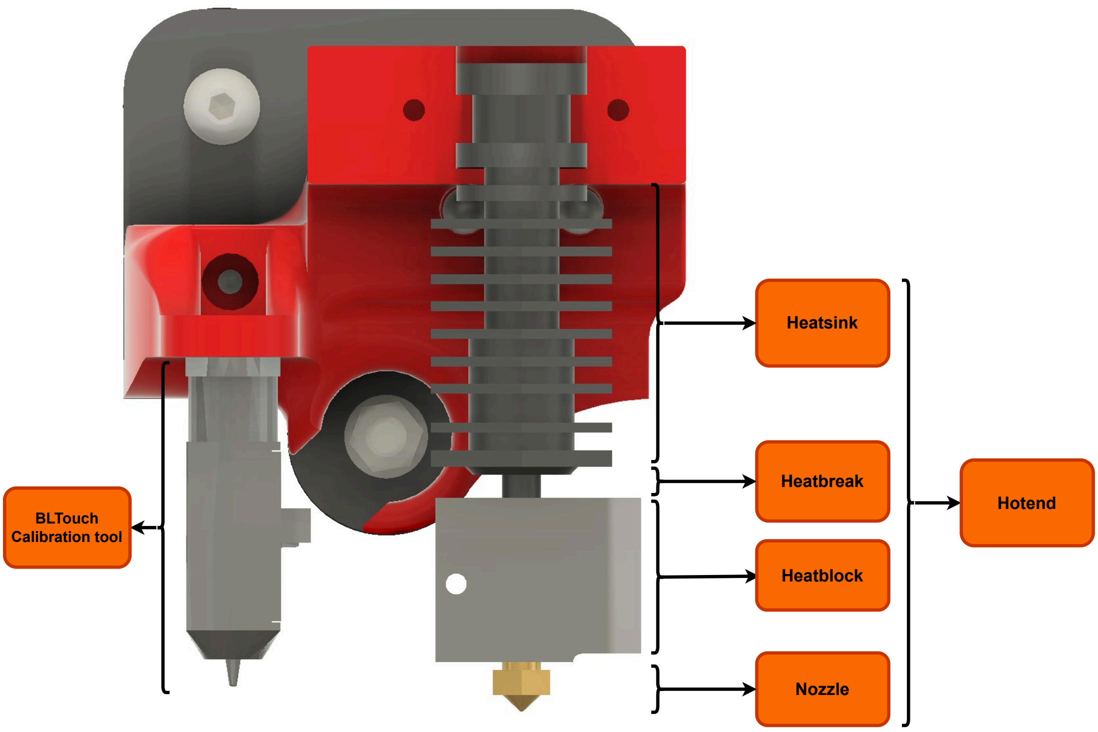


Figure 2.8: Example of a Toolhead and its components.

Recent developments on FDM printers, have permitted coupling additional components onto the toolhead, these include accelerometers which test and obtain vibration and resonances data, utilized for posterior compensation. This produces better printing quality on high speeds and accelerations. Hall sensors and inductive probes, used to calibrate the height of the nozzle relative to the printing bed, with this components instead of a static distance the entirety of the bed can be probed and have dynamic positions. Optical sensors coupled with encoders used to measure extruded filament and infer the presence of filament and existence of clogs.

The extruder handles the control of the material, namely the material flow extruded. Composed by a set of gears and a stepper motor, the extruder grabs the material and pushes it to the toolhead melt-zone to guarantee a steady flow of deposited material. Two variants for this mechanism exist, mainly differentiated by the extruder location, namely Bowden or Direct Drive.

- **Direct Drive**: The extruder is positioned directly above the printer's toolhead, which increases the weight of the toolhead component, and can be disadvantageous for certain printer configurations. However, this setup offers several advantages, including enhanced filament control due to the minimal distance between the extruder and the toolhead. This proximity allows the extruder exert a greater control over material movements and apply greater force on the material, resulting in a larger volumetric flow;

- **Bowden**: Opposed to direct drive setup, this configuration separates the extruder from the toolhead. In this case, the extruder is statically mounted on the printer structure. Between the extruder and the toolhead, a Polytetrafluoroethylene (PTFE) tube is used to create a low friction pathway for the filament to follow. This ensures that the toolhead body doesn't have additional weight brought by the extruder body facilitation movement of the toolhead, which can increase top print speed and acceleration. Due to the increased pathway between the extruder and toolhead, both filament control and total amount o volumetric flow is reduced compared to direct drive configuration.

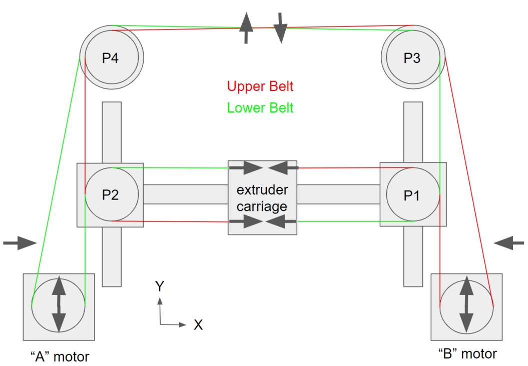
The bed or print plate is comprised of a heating element and a temperature sensor, this component is where the extruded material from the toolhead nozzle is deposited in order to create three-dimensional shape. The necessity for heated surface rises from the type of material used, certain materials must be printed and maintained at specific temperatures to properly adhere to the bed, ensuring that the printed object remains stationary in relation to the print plate throughout the printing process, preventing detachment, phenomenon often referred to as warping, and subsequent print failure.

The production of three-dimensional shapes, requires a combination of components with motion capabilities that follow a coordinate system. On some printer configurations, the bed is stationary while the toolhead is coupled on a gantry, allowing the toolhead to move along two or more axes. In others, the bed follows one axis while the toolhead moves in another. The coordination of these movements is crucial for the creation so three-dimensional structures. Movement is achieved by initially adhering to a coordinate system. The *Cartesian coordinate system* is predominantly used in FDM 3D printers, although on some printers the *Polar Coordinate System* is preferred due to their mechanical architecture. A variety of kinematic motion systems can be used, each with different advantages and disadvantages. Typical kinematic scenarios include Cartesian, CoreXY and Delta systems which are the most commonly used and will be elaborated in greater depth bellow. Other types of printers include Belt systems that take advantage of a wide mechanical belt as the printing surface; H-bot which is a form of CoreXY systems but use a single belt to move the toolhead; Polar printers are another example that differ from typical systems, these use a curved arm and a rotary build plate to print parts; and finally Scara, these use a unique combination of arms and linkages that move the toolhead [52]. The most popular movement kinematics are as follows:

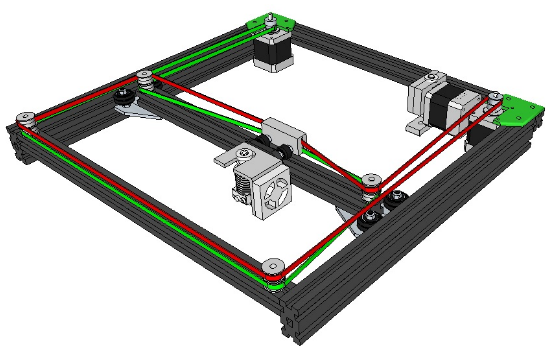
- **Cartesian Kinematics**: Also referred to as **Bed Slinger**; The printer moves the x and y axis independently from each other. Each axis has a stepper motor that confers motion, one controls movement of the printer bed plate, the other is responsible for the movement of the toolhead and the last one is responsible for the movement of the gantry in the z axis. This type of kinematic system is simple and relatively inexpensive and due to that reason the printer that uses this system are cheaper than the ones using the other systems. The system imposes limitations on the maximum speed and acceleration due to the fact that a single stepper motor is tasked with moving an independent component that often carries a significant weight.

- **Linear Delta Kinematics**: This movement system differs to the basic Cartesian Kinematics on the method to reach movement on the (x, y, z) dimensions. With a stationary bed in order to achieve movement on all three axis the printer relies on three pairs of arms that are positioned in a triangle. Additionally the arms are mounted on carriages that slide on parallel vertical linear rails. This movement changes the angle (ϕ) of each arm. The horizontal movement achieved on the printers bed is correlated to the vertical movement of the carriages making the simple application of the Pythagorean theorem [14, 23, 41]. The motion system confers these types of 3D printers with high movement velocity and faster accelerations, and on the inner bound of the printers bed, the accuracy and precision of the printer is unparalleled with other types of movements. However, due to their principle of movement some limitations are also present for example: the usage of arms to control the horizontal position introduces *Reachable area*, this refers to the area available on the horizontal positions that given a minimum angle ϕ no collisions or obstacles are present leading to a reduced actual usable horizontal area. The movement also suffers from *Effector Stability* where the vertical sliding of each arm may increase load on one of the arms resulting in an increase of imprecision's on the outer bounds of the bed. This aspect further decreases the actual usable bed size of the printer. Another negative aspect is the increased complexity of this type of printer which makes maintenance challenging and time consuming [14, 23, 52]. In view of the Linear Delta motion system these printers are substantially greater in dimensions mainly the height than other printer counterparts. The medium price range is generally higher than the other systems which is a great disadvantage for hobbyists that are looking to purchase a 3D printer.

- **CoreXY and XZ Kinematics**: The movement system also adheres to the Cartesian coordinate system; The mechanical arrangement consists of two parallel carriages on one axis connected by an additional carriage on a different axis that is coupled with the printers toolhead; This mechanism creates a gantry system; The key principle lays on how the movement the axes is achieved; On the most basic type of configuration a combination of two stationary stepper motors positioned on two opposite sides of the machine and two belts positioned on a plane and an additional system of pulleys is used to control the toolhead movement on the two chosen axis; To ensure the printer moves on (x, y, z) coordinates the remaining axis is controlled with a combination or a single stepper motor in the case of the XZ configuration [12, 70]; There are two main belt configurations that achieve this type of movement system, both use two belts. The difference lies on the position of the two belts, where one the belts have an offset in relationship with each other; the second most important type, the belts form a cross pattern [12, 60]. This difference is depicted on Figure 2.9. CoreXY systems have great advantages such as movement speeds and accelerations, although lower than Delta printers it is much greater than Cartesian Kinematics type of printers, Core printers ensure equal quality across the entire bed and consistent performance. Additionally, the system ensures a greater rigidity reducing vibrations on the structure that contributes to print quality and due to the motion system it handles greater loads and generates more torque than the other kinematic types, useful for when the toolhead has an increased weight [52]. Furthermore this system type has been adapted to handle even higher speeds and accelerations with the introduction of **Hybrid Core XY** that makes use of four stepper motors and additional belts to overcome the gantry inertia and weight restrictions;



(a) Cross Pattern belt path (gathered from [60]).



(b) Offset belt path (gathered from [60])

Figure 2.9: Two different belt configurations for CoreXY kinematic system.

2.4 Artificial Intelligence

Artificial Intelligence (AI) is a broad concept that often refers to, in simple terms, the mimicking of humans cognitive learning methods and the ability to perform tasks with the acquired knowledge. Humans attain insights of concepts and principles through the aid of real-life examples that help them achieve comprehensive understanding of a subject. Similarly, AI comes as a solution for computers to learn and perform tasks in a manner that replicates human actions without the need of coding every scenario and concept [5, 31].The progress in this field id closely linked with I4.0 standards, providing add cost-effective solutions that enhance and automate tasks while increasing productivity. Accompanied the development of highly sophisticated algorithms in recent years has brought considerable attention to the field making it as a highly promising tool to create intelligent systems. Despite the high regard for this field, several challenges remain predominant on its implementation, as highly skilled workers are necessary to adapt and develop machine learning models suited for the task at hand. Additionally, the quantity and quality of data is of utmost importance as variations or incorrect data can hinder the training phase of a model and result in its inability to infer correct decisions [49, 31, 33, 5]. The application of this technology is widespread, fields such as Healthcare, Industry, Automotive, Economy, already employ AI based systems to increase their productivity.

Machine Learning (ML) is a subset of AI, that is commonly used interchangeably, it refers to the employed methods that confer systems with a degree of intelligence, it encompasses all technologies and algorithms such as *Support Vector Machines (SVM)*, *Decision Tree*, *K-Nearest Neighbor (KNN)*, utilized on systems that are independently capable of learning and improve themselves from extensive quantities of data which are later used to solve specific tasks without or with minimal human intervention [31]. The application of ML algorithms entails an initial training phase to create a model, on this step the model iteratively learns from data that depicts a specific and independently learns insights and establishes complex relationships and patterns that may be arise from data. The precision of decisions made from ML models heavily relies on the available quantity and quality of data [33, 31, 5].

Until recent years, acquiring large quantities of data posed a significant challenge for researchers, as it is essential for the model training and benchmarking phases. Image recognition tasks, require large scale datasets such as *ImageNet*, composed of more than 14 million labeled images; and Cannadian Institute For Advanced Research (CIFAR) *CIFAR-10* which includes 80 million tiny images. These are two available examples of data sources that enable researchers to create and benchmark new models which require vast amounts of images. Furthermore, ImageNet in an attempt to create a more competitive environment, created the *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)* competition where researchers can submit models for evaluation that establish additional benchmarks for newly developed architectures [28, 39, 63, 38, 9].

Depending on the task, machine learning algorithms can be sub-categorized into three different classes.

Supervised Machine Learning refers to algorithms that in order to learn complex relationships between data, require data to be labeled *a priori*. Every element within the training dataset must be labeled with its ground truth, ensuring that each element is accurately identified according to what it signifies. This labeling is often achieved by extensive human intervention. After the initial training phase, the trained model can process unlabeled data to test the generation of valid decisions and categorizations. Classification and regression problems are two types of tasks where supervised machine learning methods excel at [31, 72, 34, 15]. Some notable examples that fall under this category include [31, 72, 34]:

- **Support Vector Machines** (SVM): This algorithm, depicted on Figure 2.10(a), is based on the calculation of margins. SVM is optimal for regression and classification tasks that need separation of classes. Given a labeled dataset, the method separates data into two different classes creating a hyperplane between them. To reduce classification error, the margins between data are also adjusted to fit the labeled classes.

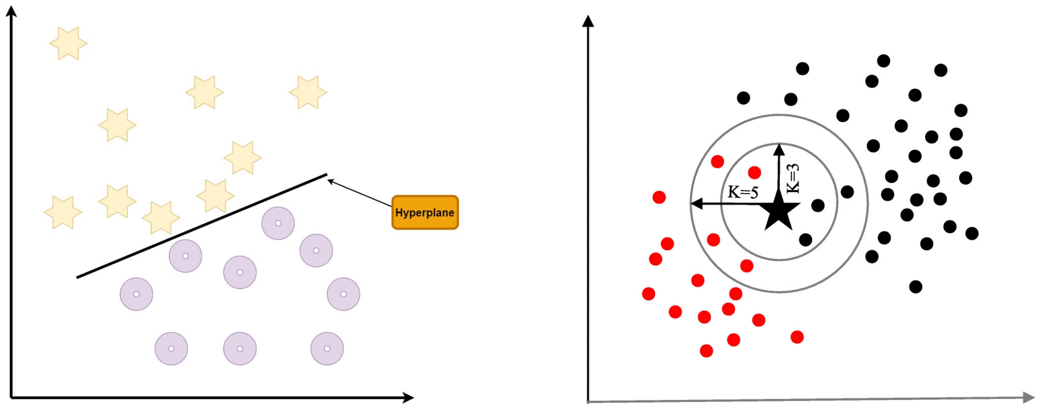
- **K-Nearest Neighbor** (KNN): One of the oldest algorithms suitable for classification and regression purposes, Figure 2.10(b). For classification the idea behind the algorithm lies on considering K neighbors to take a take a majority 'vote' around a certain point. The label exceeds a given percentage of votes sets the label classification for that point [80].

- **Naive Bayes**: Employs a classification technique based on the Bayes' probability theorem, Equation 2.1. Considers that events are independent from each other but attributes that are part of a event can be interdependent with others [17, 72].

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.1}$$

Unsupervised Machine Learning is another class of algorithms that, opposed to supervised machine learning methods, does not require labeled training data, in this type of algorithm, unlabeled or unstructured data is accounted for. Mitigating time-consuming processes required for labeling datasets. Conversely to supervised machine learning, these algorithms learn and detect patterns on their own, gathering knowledge by establishing relationships when analyzing through the data. *K-Means Clustering* is one example of unsupervised machine learning algorithms. One important consideration with this type of learning method is the requirement for a substantial amount of examples in the dataset that enable model training, without these [49, 31].

Reinforcement Learning consists of algorithms that learn trough a reward system. By receiving feedback from predictions that are correct or incorrect the model can adjust



(a) Illustration of SVM algorithm (adapted from [72]).

(b) Illustration of KNN algorithm (gathered from [72]).

Figure 2.10: Illustrations that depict SVM and KNN algorithms [72].

itself to increase its precision. When a correct prediction is made feedback is given to the model its weights are re-calibrated, when an incorrect inference is made a reprimand is therefore given and weights are adjusted so not to make the same incorrect prediction [31, 48].

Transfer Learning comes as solution to circumvent arduous process that involve training a model from scratch. The key word for this method is re-use. This technique leverages knowledge from previously trained models on larger datasets such as the aforementioned ImageNet, and utilize it as the base knowledge for another model. The new model uses this previously gained knowledge and only the last few layer need to be trained with new problem specific data. It aims at reducing model training time by using the knowledge of one previously trained model, it has shown excellent performance on limited datasets [48, 36, 29].

2.4.1 Deep Learning

Deep Learning (DL) is a type of machine learning that belongs to a broader concept referred to as *Artificial Neural Networks (ANN)*, inspired on discoveries related to neuroscience findings and human brain functions. It comes as the next-level AI method that intends to close the gap between the human brain and intelligent systems, providing an architectural network that mimics synaptic connections between neurons present on the human brain. The basic building block on this type of architecture is a novel processing unit for neural networks , referred to as the *perceptron*, brought to us by Frank Rosenblatt in 1957. The perceptron, Figure 2.11, sets on using a mix of binary inputs $x_1, x_2, x_3, ..., x_n$ with their associated configurable *weights*, $w_{k1}, w_{k2}, w_{k3}, ..., w_{kn}$. This novel processing unit produces a weighted sum of all inputs and multiplies it by their correspondent weight representing their importance. After this weighted sum, the constant *bias* threshold is added to allow the activation function to shift. The activation function then receives the summation of the weighted sum and the bias to normalize the output [19, 45, 6, 80, 15].

The perceptron is the base for feedforward neural networks and Multi-layer Perceptrons (MLP) which are mainly used for linear problems. In real world scenarios most problems are nonlinear, this is where Deep learning comes in [15, 80].

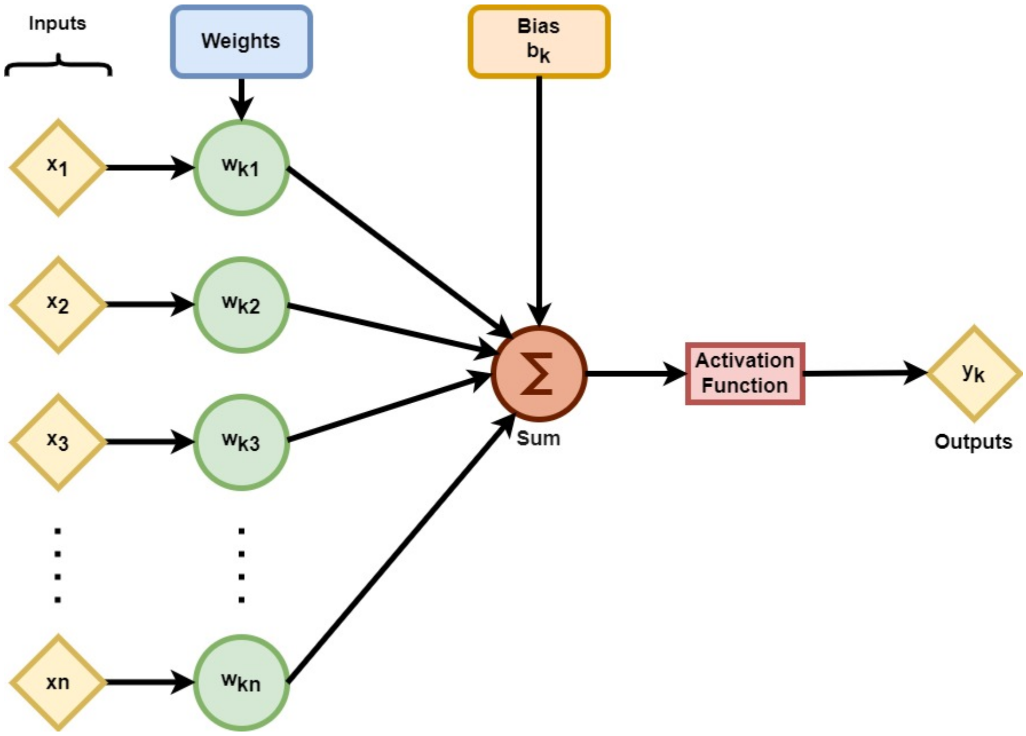


Figure 2.11: Illustration of a Perceptron (adapted from [45]).

The activation function is chosen according to the desired output, which can either be linear or non-linear. This last step confers the perceptron with a slower learning phase to increase precision. Some notable activation functions include [45, 6, 15, 51, 50, 59]:

- **Sigmoid Function** - Also called Logistic Function, mostly used for logistic regression and Multi-layer neural networks on non-linear problems normalizing the output in the range of $[0; 1]$. Computational expensive and presents Vanishing Gradients problem. On the scope of sigmoid functions three variations exist Sigmoid-Weighed Linear Units, Derivative Sigmoid-Weighted Linear Units and lastly Hard Sigmoid function.

$$sigmoid(v) = \frac{1}{1 + e^{-v}} \tag{2.2}$$

- **Rectified Linear Unit Function (ReLU)** - A popular activation function due to its simplicity, computational efficiency and the ability to speed up training stages of a model, additionally, it alleviates vanishing gradients problem. Conversely, due to having a non-zero mean activation, bias shift problems can occur that. When the bias shifts towards a negative value the neuron might never active and henceforth become "dead". Variations of this function include Double ReLU, LReLU, PiLU and

PReLU [30, 10].

$$RElU(v) = max(0, v) = \begin{cases} v_i & \text{if } v_i \geq 0 \\ 0 & \text{if } v_i < 0 \end{cases} \tag{2.3}$$

- **Leaky ReLU** - Comes as one of the answers to reduce the "dying neuron" problem that the ReLU activation function introduces. Exhibits the same benefits as ReLU but requires some more fine-tuning that when improperly done can cause activation problems [10].

$$LeakyRElU(v) = max(0.01v, v) \begin{cases} v & \text{if } v \geq 0 \\ 0.01v & \text{if } v < 0 \end{cases} \tag{2.4}$$

- **Hyperbolic Tangent Function (Tanh)** - Normalizes the output values between $[-1; 1]$. Commonly used for multi-layer neural networks. Enables stable training processes compared to sigmoid. Suffers from the problem of vanishing gradients and is computationally expensive.

$$Tanh(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}} = \frac{\cosh(v)}{\sinh(v)} \tag{2.5}$$

- **Sigmoid Linear Unit Function (SiLU)** - Also known as Swish function. Shows better performance when training deep models. Ramachandran, Zoph, and Le [59] state that "Swish consistently matches or outperforms ReLU on deep networks applied to a variety of challenging domains such as image classification and machine translation.".

$$silu(v) = v * sigmoid(\beta v) \tag{2.6}$$

β is a constant or learnable parameter.

- **Softmax Function** - Works with problems that require multi-class classification, it works by assigning probabilities to each class type. The sum of all probabilities equals to 1 and normalizes the output values in a range between $[0; 1]$.

$$softmax(v) = \frac{e^{v_i}}{\sum_{j=1}^{n} e^{v_j}} \tag{2.7}$$

Processing done by the activation function is made so not to exceed a configured threshold, information passed to the next perceptron is then limited by this threshold. Deep learning networks are based on the principle set by ANN, in this case, a variable amount of perceptrons, or neurons, are interconnected creating groups, which then create vast networks of interconnected layers [45, 80].

The complexity and quantity of hidden layers within this layered architecture are crucial for feature extraction from the input data. As the number of layers increases, so does the network's level of abstraction. In image-related problems, groups of layers can be assigned with distinct functions. Some layer groups may focus on identifying regions of interest within an image, while others are employed for extracting features to ascertain the content of the image [31, 86].

Hyperparameters are crucial settings determined prior to the learning phase of a model, including the number of layers and neurons, initial learning rate, and the choice of activation functions. These configurable elements are highly customizable during the construction of a DL network and are not configurable during the model's learning stage, therefor a methodical trial-and-error approach is necessary to fine-tune the models for optimal performance [86, 31, 45].

Furthermore, for the network to effectively learn from the analysis input data it relies on *backpropagation* algorithms that recalculate weights, which dictate the importance of a given feature, according to the error produced on the output prediction and the actual output. Different algorithms have emerged in recent years, including *Recurrent Neural Networks (RNN)*, mostly used for language processing and time-series data, relies on feedback loops that sequentially learns patterns; *Autoencoder*, used for natural language processing, consists of encoding the inputs into a two-dimensional layer and final decoding stage that reconstructs the original input with the learned features [31, 86, 45, 19, 80].

Another honorable mention is *Convolutional Neural Networks* (CNN) which is a type of complex neural networks that excel on image and video recognition tasks, these algorithms will be explained in further detail on Section 2.4.2 due to their importance to the current work.

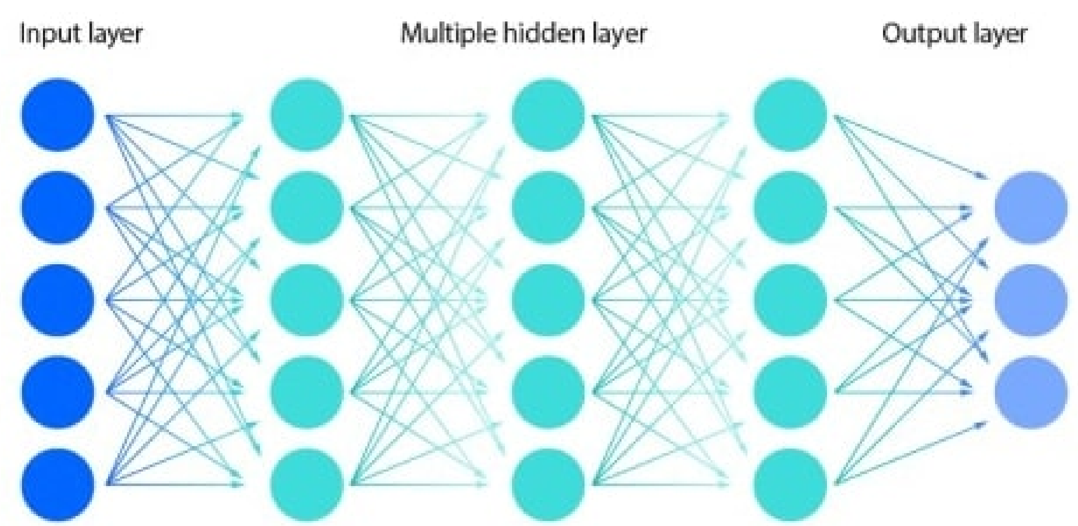


Figure 2.12: Deep Learning Network (adapted from [80]).

2.4.2 Computer Vision

As an application of machine learning algorithms, enabled by the development of deep learning architectures, computer vision involves the use of deep neural network systems capable of analyzing visual data. This class of machines learning aims to emulate human visual perception. Computer vision algorithms provide excellent solution for tasks such as predictive maintenance, three-dimensional visual inspection, preventive safety technologies, which are already being employed in manufacturing industries and autonomous vehicles with crash preventive technologies [15, 62, 11].

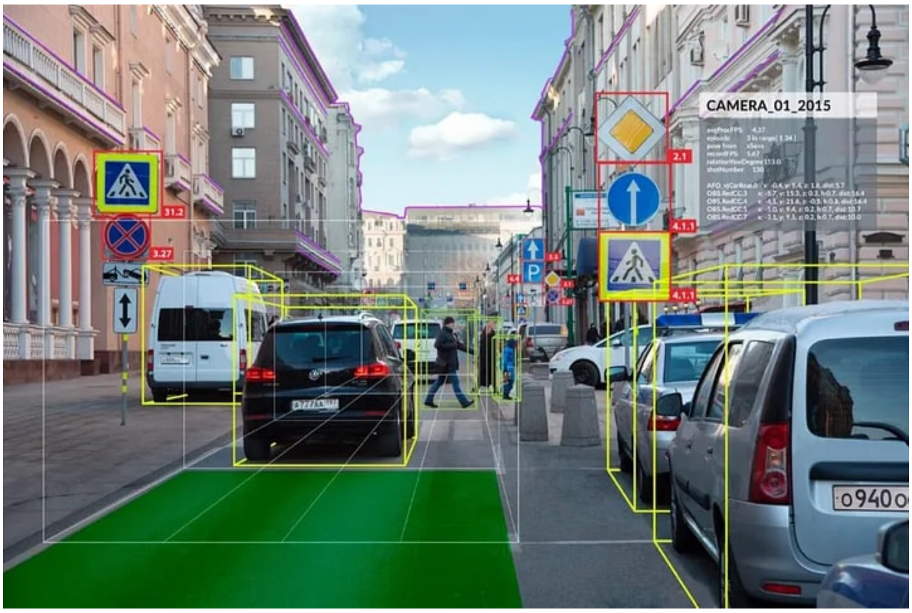


Figure 2.13: Representation of computer vision algorithm that recognizes road signals and vehicles (gathered from [11]).

As previously mentioned Convolutional Neural Networks (CNN) are a class of neural networks that are particularly well suited for tasks that involve pattern recognition and establish spatial relationships as is the case of image and video data where information is organized into rows and columns. This architecture differs from typical ANNs because it arranges neurons in three dimensions, enabling the processing of inputs with spatial attributes such as height, width, and depth. CNNs are structured such that the initial layers capture fundamental features such as edges and corners. As the input data moves deeper into the network, it extracts progressively more complex features. This type of algorithm architecture generally consist of three primary components: the convolutional layer, the pooling layer and the final fully connected layer [4, 31, 54, 19].

The **convolutional layer** refers to the process of applying a filter or *kernel* - essentially matrices with small spatial dimensions but with the same depth as the input - across all pixels of an image. As the kernel moves across each pixel, it performs a scalar product calculation based on the filter's dimensions, resulting in the creation of a feature map that highlights important patterns present on the input. This layer also plays a pivotal role in reducing the complexity of the model. Three hyperparameters can be adjusted: the depth of the output, the stride and the padding. Adjusting the depth of the output can reduce the number of neurons in a layers, which in turn reduces the pattern recognition capabilities of the model. The stride controls how the kernel passes through the input; a value greater than one will result in pixel skipping, which reduces the spatial dimension of the output feature map and downsamples the input. Finally, padding involves adding additional pixels to the border of the input, affecting the spatial dimension of the output feature map [54, 19].

Following a convolutional layers is the **pooling layer**, which is responsible for reducing the spatial dimensionality of the features maps captured in the previous step without losing information. *Max-pooling* is a common pooling operation where a kernel with dimensions of 2x2 and stride of 2 is applied on the output of the convolutional layer effectively reducing the dimensions in half. Generally, kernels with dimensions above 3 will greatly erode the models ability to recognize patters [54, 31].

The **fully connected layer** is generally positioned as the last layer. It employs a classifier, such as the Softmax function, to evaluate the feature maps captured by the preceding layers [4, 54, 45].

2.5 Anomaly Detection

The requirements set by I4.0 can be considered as an a accelerator for Additive Manufacturing technologies, Section 2.2. The fields growth is not only attributed to companies, but also to open-source communities and individuals that continually develop and perfect new components and 3D printing architectures with the goal of conferring reliability, decreased printing times and material wastage, while at the same time achieving these goals without sacrificing on *quality*. In the current scenario, quality refers to the aesthetic and mechanical dimensionality that the finished or ongoing printing object exhibits in comparison with the intended outcome, these traits directly relate to the chosen printing parameters during the slicing stage of the model, Section 2.3, including layer height, extruder temperature, bed temperature, flow rate and feed rate.

Currently, quality assessment heavily depends on human visual inspection, which can be time-consuming, destructive, and sometimes inaccurate. The inability of a human operator to consistently monitor a printing job highlights the necessity for systems capable of detecting anomalies and initiating actions that save materials and reduce printing times.

When considering FDM 3D printing machines, as discussed in Section 2.3, the technological field has experienced significant advancements in a short time span. These advancements include probing sensors that level the printer toolhead relative to the bed plate, ensuring consistent layer height across the entire bed. Additionally, sensors that detect material runout during a printing job can pause the ongoing print, allowing for filament changes and resumption of the print, this aspect confers additional reliability to the technology by reducing wastage and introducing a recovery plan in the case of material depletion.

Sensor-based solutions are relatively inexpensive, yet they necessitate additional components for synchronous interaction. Moreover, not all printers can be directly upgraded, often requiring extra hardware components such as a new controller board and modification to existing components. The labor involved in retrofitting a 3D printer may not always be universally feasible.

Other modern mechanisms leverage cameras coupled with computer vision models, Section 3.5, running on a Single Board Computer (SBC) or other hardware, to assess incongruities in printed structures by capturing images of printed 3D objects. These systems offer a robust alternative to retrofitting a 3D printer. Although additional hardware is required, in some cases, there is no need to alter the machine. Furthermore, for consistent monitoring during an ongoing print job, using a camera to evaluate the condition of the printed structure can reduce the need for human operators.

Considering the objective outlined in Section 1.2, the subsequent paragraphs aim to reveal existing systems capable of anomaly detection on AM 3D.

2.5.1 Tranditional Algorithms

Delli and Chang [13] 2018, conducted a research which aimed at using supervised machine learning algorithms to monitor 3D prints and thus producing a real-time anomaly detector. The described system classifies prints as 'good' or 'defective', if the latter is ever encountered the printing process can be stopped. To this end the authors took advantage of Support Vector Machines (SVM). The proposed system defines checkpoints where the printer pauses and captures a snapshot of the printing part on the vertical axis. As a result of this capturing technique, this captured image is only able to detect surface defects. The checkpoints were selected by the geometry of the 3D printing part. The authors also used different types of filaments, such as PLA and ABS due to their different properties. The SVM model was trained with a labeled dataset containing both 'good' and 'defective' classes. After successfully training the model, the SVM can be used to classify any other input data. The model was developed using Python with the help of Scikit-learn library. The authors were able to create a real-time fault detection system that could detect surface errors and if the printer had a clogged nozzle. Several drawbacks were also highlighted. Although the objective was successful, the detection mechanism has its limitations. It can only evaluate the surface of the print, so errors on other axes are not considered in this system. Another notable drawback is that the printer needs to pause to capture an image, which is particularly detrimental to the rapid prototyping principle in AM technologies as it introduces longer printing times. Proposed improvements include the usage of a nozzle camera, providing additional coverage of the printed object on every axis.

Adding to the previous method, Gobert et al. [22] 2018, proposed a system that similarly leverages SVM algorithms. The study aimed at detecting porosity anomalies in powder bed fusion machines, disclosed on Section 2.2. Powder bed fusion machines use high energy laser to solidify layers of material. A common challenge in this technology is the fusion of layers, as contamination between layers can occur, leading to incomplete bonding and porosity between the layers. The authors proposed a monitoring system that could detect layer wise anomalies using imaging. The training phase required manual labeling of the dataset, the chosen labels were 'anomalous' and 'nominal'. The authors produced an effective classifier, reaching 85% accuracy. Improvements denoted by the researchers to increase the model's classification performance include utilizing higher image resolution, better lighting and additional cameras.

Additionally, Aminzadeh and Kurfess [3] 2019, noted that metal powder-bed AM processes face intrinsic challenges with layer fusion, as there is a high likelihood of fragmentation or defects in the fusion, resulting in poor geometrical accuracy. The authors aimed at solving the same defect as previous research. To address the issue highlighted the author proposed an online real-time monitoring system that inspected defects such as porosity and measured the grade of fusion every layer. A Bayesian classifier was utilized in this system, the training dataset was labeled binary "good" or "bad" classes. To achieve best performance, the classifier was built with five frequency domains which resulted on 89% accuracy when identifying parts as defective or not. In the end of this research the authors identified some challenges including the quality of the captured images denoting that images should be captured with higher resolution, and the importance of a good training dataset.

Kadam et al. [34] 2021, offered a comparative study on pre-trained machine learning algorithms, the case study presented performances of AlexNet, GoogLeNet, Resnet18, Resnet50 and Efficientnet-b0 models when applied to feature extraction. The study was created to detect anomalies on FDM machines eight potential anomalies identified including 'Warping', 'Blistering', 'Porosity', 'Cracking', 'Residual stress', 'Poor surface finish', 'Stringing' and 'Material shrinkage'. For training and classification, the authors used different types of algorithms including SVM, K-Nearest Neighbor, Random Forest, Decision Tree and Naive Bayes, the produced classification was desired to give either "good" or "bad" part. The case study was successful, the authors concluded that the combination of AlexNet with SVM learning technique produced the best outcome with 99.70% accuracy. The system could be used for real-time layer monitoring, as the best combinations also demonstrated to be less computationally expansive.

The methods above explore the application of traditional supervised machine learning methods to achieve monitoring capabilities in various types of 3D prints, yielding comprehensive and successful results. Despite the authors' abilities to produce respectable outcomes, a common challenge found lies on the creation of a good dataset for the training phase of the model. The need for a labeled dataset is evident. Furthermore, all classifiers were limited to produce binary decision, such as categorizing prints as 'good' or 'bad', this aspect eliminates the possibility of producing detailed insights about the type of problem that the system detected. The created systems offer valuable information and prove advantageous to the printing process, yet it's evident that there is room for improvement.

2.5.2 Convolutional Neural Networks

Krizhevsky, Sutskever, and Hinton [39] 2017, wanting to improve the used object recognition methodology at the time, the authors created a deep CNN later called AlexNet. Image recognition was based on labeled datasets that could be described as small, and these recognition tasks performed successfully. A problems with this type of approach comes from objects which exhibit various characteristics that could only be correctly identified with massive datasets (in the order of millions of elements). The outcome was the creation of a convolutional neural network that took advantage of the ImageNet database to compensate for lack of available data. This CNN is composed of 5 convolutional layers, 3 fully-connected layers and SoftMax function. The produced CNN was groundbreaking and topic to many researchers, although the authors used supervised machine learning method to train the CNN, they make space for improvement stating that unsupervised pre-training should prove to be an advantage.

Scime and Beuth [67] 2018 , realized that powder spread 3D printers offered visualization for the printing part, but lacked capabilities as inspection of the part. The authors proposed a real-time monitoring system with the help of computer vision that detected anomalies in the 3D printed structures. The developed system could detect and classify errors in real-time, during the printing process. The authors labeled six different types of anomalies that suggested problems in the printing part. The system used a camera on top of the 3D printer to acquire visual information, the images could not be utilized directly, to solve this issue the images were sub divided into regions. For the model training, the system used machine learning method called Bag-of-Words (BoW), which resulted in extensive human interaction, each image was manually labeled in one of the six proposed identified anomalies and stored into a database. The authors pointed that the more image patches the database possessed, the better the algorithm performs. Overall performance was divided into three different metrics, the system could correctly detect if no anomaly was found in 100% of the cases, when the anomaly is found the system could only accurately detect the type in 95% of the cases, and 89% classify that an anomaly is present but not classify it.

In the same year, the previous work was expanded by Scime and Beuth [66] 2018, the authors proposed multi-scale CNN integration to increase the classification accuracy of the previous work. The system was also capable of classifying six types of anomalies as the previous method. The usage of supervised machine learning method BoW was a limiting factor as it relied too much on human interaction, so a new approach was developed. The prior work was outdated by a new revolutionizing strategy, AlexNet, the new method proposed using this CNN in conjunction with transfer learning. The multi-scale aspect comes from the fact that the AlexNet CNN input images are in color and with 227x227 pixel dimensions, the proposed system only captures grayscale image patches of different sizes. The multi-scale aspect comes from the division of patches into three different levels according to their sub-region: level 1, 25x25 pixels; level 2, 100x100; level3, 900x900 pixels that represents the entire captured image which is then rezised to 200x200 pixels using bicubic interpolations. To solve the color format input and size needed for AlexNet, the proposed method chose to utilize the different patch levels as the color channels and resize each patch to the required 227x227 pixel size using bilinear interpolation. The main advantage of AlexNet comes from its outstanding backpropagation algorithm that correctly chooses the kernel weights, marking it a critical element for accurate classification. During the training phase the backpropagation proved to be a resource-intensive task. To solve this problem with the multi-scale CNN, the authors leveraged transfer learning which allowed the system to use a previously trained AlexNet with already calculated initial weights. After the creation of the multi-scale CNN, the authors compared the results with the previous study, which used BoW to recognize anomalies, AlexNet CNN trained with only the level 1 patches referenced before and the built multi-scale CNN. The results showed that the newly developed multi-scale CNN had higher performance in recognizing anomalies over the other two methods, in 94.2% of the cases the model could detect part damage anomalies.

Another proposed work, Kwon et al. [40] 2020, saw equal potential on using deep learning neural networks, and proposed case study that could detect anomalies with the help of machine vision. The created system was composed of a high-speed camera and a deep neural network to classify the melt-pool images with laser power. Pre-processing was done to the images that was comprised of a noise filter. The neural network was constructed with a variable hidden layer number. The model was not used for regression because of weak performance but it was able to detect anomalies with successful results.

Kim, Lee, and Ahn [36] 2022, proposed a system to accurately detect spaghetti-shape anomalies in FDM 3D prints. This defect occurs when the deposited filament fails to bind to already deposited layers, making the printer extrude filament into the air thus creating a spaghetti-shape error. Due to absence of large datasets, the authors proposed using an initial small dataset to train a base model, and incrementally deep transfer its knowledge to future models training using CNN. The training and validation dataset was comprised of defective images exhibiting spaghetti-shape anomalies and due to its rather small size, the authors recurred to data augmentation techniques considering the geometrical shape of the object to provide more examples to the model. The original dataset was comprised of 327 images with both good results and spaghetti-shape errors, the augmented dataset resulted in 14,635 images with both classes. The usage of pre-trained models such as ImageNet, recurring to these pre-trained models that already have knowledge prove to be an efficient way to recognize new objects. The authors recurred to four popular CNNs pre-trained on ImageNet including VGGNet, GoogLeNet, ResNet and EfficientNet. Proven to be unfeasible due to impossibility of directly using a trained CNN as the basis for the next one due to class limitations, techniques for CNN weight extraction as Fixed feature extraction and fine-tuning, the last was chosen. The study produced successful results, image augmentation proved to be a viable option for increasing the data diversity on the original dataset. The authors concluded that VGGNet gave the best accuracy when identifying spaghetti-shape anomalies with a result of 94.2%. Furthermore, the best overall performance was achieved by ResNet50 when applied to devices with limited computational specifications, the CNN managed to achieve 91.4% accuracy with low computational taking 0.58 seconds when the system is applied to the Raspberry pi 4B SBC.

Westphal and Seitz [78] 2021, developed yet another interesting system for SLS machines. The authors used deep CNNs including VGG16 and Xception, pre-trained models with data from ImageNet, as the basis of their work. The purpose of the study is to increase the level of automation in this type of production systems. The authors highlighted some anomalies that could detrimentally non-functional 3D objects. For image acquisition the system utilized a two camera setup consisting of a relatively low-cost cameras that could record high quality video, these were connected to a Raspberry pi 3B+ SBC and a touchscreen; the mentioned camera was meant for monitoring purposes. Another camera positioned on a tripod capture images of the SLS machines display.

The images were analyzed with two-stage CNN, this entails that the first stage CNN is used to locate the defect and the second stage CNN handled error masks for the location, the authors affirm a great improvement in error detection accuracy using two stage CNN architecture. The created dataset was also specifically designed to handle error cases, no further details were disclosed concerning the accuracy of the two-stage CNN on the custom-made dataset. Information about the dataset creation was properly disclosed, built using the monitoring camera, every 20 frames the 20th frame was saved as a JPEG image and compressed. The resulting dataset was constituted by 9426 images of the bed. The image labeling consisted of the binary 'OK' with 7808 examples and 'DEF' with 706. Due to the discrepancy in the number of examples between the two classes an imbalance problem is evident. To solve this problem the authors repeatedly resort to various techniques such as increasing the model sensitivity to the low volume class. Additionally, as in other literature, data augmentation was performed with operations such as re-scaling, horizontal flip, zoom, shear operation and random rotations. Transfer learning was also used for the two-stage CNN training and Fine-tuning was used for this purpose.

The implementation used the Framework TensorFlow on python programming language. The case study produced favorable results; a real-time monitoring system was successfully built. VGG16 CNN was able accurately detect defective parts, with 95.8% accuracy and 93.9% precision. Comparing VGG16 and Xception results, resulted in an out-performance from VGG16 with data augmentation. For future work the authors proposed studying the performance of CNN with even smaller datasets, and the effect of pre-processing on the performance of the models. Furthermore, the creation of a custom dataset with more classes was recommended to detect other anomalies.

Jin, Zhang, and Gu [32] 2020, developed a method based on computer vision to detect and predict inter-layer anomalies on fused deposition modeling 3D printers. The authors used a camera based system with deep learning techniques to monitor the printing process. The proposed system was made with the intention of determining anomalies in the delamination class. These anomalies refer to an offset between the printer nozzle and the print, the result of this offset is irregular layer adhesion making the printed part unusable. The camera setup made in this case study is similar to the one proposed by the present document, the position of the camera was mounted in order to focus on the printer nozzle. For error classification the proposed classes were ”High+”, ”High”, ”Good” and ”Low”. These categories refer to the offset between the nozzle and the print. In the image acquisition stage four prints were made with the specified anomalies classes, resulting in 900 captured images each print.

The authors also recurred to image augmentation techniques, for each image four others are produced. For the system to correctly determine anomalies, the convolutional neural network ResNet algorithm was used has the deep learning method. The authors modified the ResNet CNN by removing the last layer and adding two fully-connected layers for the output to be a four element vector. The system was able to produce high precision on ”High+” and ”High” but the other two classes fell short in accuracy confidence. Improvements were made to further improve the CNN training, a new class ”Null” was identified to resolve issues where the nozzle introduced discrepancies on the CNN classification. This approach was successful and increased the model's testing accuracy from 95.5% to 97.8%. The authors also proposed a method to detect warping anomalies that was based on strain-gauges. This method also produced favorable results; the system could detect when warping anomaly was happening. In the end of the literature there was space for future work, the authors intended to expand their work by creating an autocorrection mechanism when delamination anomalies are detected, this recommended work is very related to the proposed system of the present document. Furthermore, infra-red cameras were also recommended as a data acquisition on future work.

The aforementioned systems leverage CNN architectures to analyze image data, proving to be an effective method for enhancing performance when dealing with this type of data. The proposed systems demonstrate excellent results in anomaly detection, providing additional information about the encountered error types compared to the more traditional methods previously discussed. Furthermore, the ability to use transfer learning methods in some systems, pretrained on larger datasets, mitigate the need for extremely large datasets with specific images of anomalies.

A vital aspect not found in prior systems is the interrelation of anomalies. When considering FDM 3D printers, anomalies that are visually identifiable can significantly impact the printing process and the quality of the final product. Some anomalies can be misdiagnosed as the root cause, this issue is significant because different anomalies require corrective actions.

Brion and Pattinson [7] 2022, identified that the current automated anomaly detection systems could not be generalized across different printer types. The authors proposed a multi-head neural network architecture that could automatically evaluate images dubbed Collaborative Autonomous Extrusion Network (CAXTON). A nozzle camera was utilized to capture material deposition snapshots in junction with a raspberry pi 4 SBC.

The authors stated that, although error detection has proven to be an essential tool in AM processes, to further develop the technology, anomaly detection systems should be coupled with active error correction. The authors proposed a automatic image labeling system, developed to avoid time-consuming human interactions required to manually label the dataset. This resulted on printing 192 objects for data collection, where a range of data parameters from the ongoing prints where captured and recorded into a .csv file along side a snapshot. To automatically assign labels to the specific problems is done by calculating the deviation of the captured printing parameters from the ground truth. Four different anomalies were selected as they are interdependent and can be misdiagnosed with each other, these include flow rate, feed rate, z offset and hotend temperature, each anomaly labeled as either ”low”, ”good” or ”high”. CAXTON network consists of a single backbone capable of determining multiple labels for each anomaly, developed recurring to python and using the PyTorch Framework.

The training process was split into three different stages and used transfer learning to create a consistent model. The correction system in this approach was tested on various gcode models. The user had to select a pixel region around the nozzle tip, this region was then used for the detection phase. The system implementation was successful, leading to a reduction in time-consuming human interactions. However, some post-processing was performed by the authors to fine-tune the dataset and prepare it for model training. The developed model could detect all anomalies classes and attribute the specified label to each, achieving accuracies for flow rate, feed rate, z offset and hotend temperature of 87.1%, 86.4%, 85.5% and 78.3% respectively.

The authors note that there is still room for improvement, include testing the model on different brand 3D printers, this suggestion comes as to expand generalization and asserting the technology. The approach was able to detect four different types of problems but being closed-loop (the system has no interaction with the printers’ mechanical properties and sensors) information about mechanical failures was not considered, taking this data from the printer could increase error categories and could provide an even greater error logging. Another recommendation is model fine-tuning the network and usage of transfer learning for small datasets, to solve problems such as sporadic inconsistencies and environment errors.

2.5.3 Summary

The exposed systems prove to be advantageous in enhancing AM technologies by creating useful tools that enable a degree of autonomy to 3D printing machines, hindering the need for human monitoring. Systems that utilized traditional supervised machines learning algorithms produced good results but lacked both performance and the ability to handle cases where a binary decision is insufficient. In contrast, architectures that leverage deep CNNs demonstrated significant advantages when using imaging data. Performance increased compared to traditional algorithms, and a wider range of anomalies could be considered. Additionally, models in which a singular anomaly was considered, the systems were able to provided detailed information about the anomaly. The final architecture is particularly interesting as it not only considers different anomalies but also the interdependencies between them, which proves to be a clear a advantage over other systems. Additionally, it provides information about each type of anomaly. The developed system also exhibits a higher degree of autonomy compared to other research, as it can be generalized to work with other machines.

3

System Architecture

The current chapter discloses fundamental concepts and technologies that support the development of the system highlighted on Section 1.2. It starts with an introduction, followed by common print anomalies in Fused Deposition Modeling (FDM) Three dimensional (3D) printers, types of filaments, and system requirements. It is then explained in great detail all information about the proposed modifications to the anomaly detection system in section 3.5. The chapter ends with a final section which explores the development of the proposed dataset, with detailed information about its implementation.

3.1 Introduction

Fused deposition modeling 3D printers, as explained on Section 2.3, produce three dimensional structures by depositing melted plastic. The intricate process of 3D printing is the result of various components working in harmony. Each element plays a crucial role on determining the final quality of the printed part. Every step that composes 3D printing an object is crucial, from the chosen printer configuration, to the slicing stage where parameters are adjusted according to the used filament and the printer setup. Mechanically, the precision and torque of the stepper motors that control each axis, the extrusion consistency, and the stability of the bed plate, all contribute to the accuracy and detail of which an object is printed. Moreover, the type of filament and even brand can significantly affect the strength, flexibility and appearance of the object.

Optimizing this parameters is crucial, as evidenced by research exposed on Section 2.5, the application of analysis of variance (ANOVA) methods, as detailed in the article Espino et al. [18] to control printing parameters for optimal quality. During the slicing stage, parameters such as infill pattern, infill percentage, extrusion flow, amount of wall, build orientation can significantly alter the mechanical properties, dimensional accuracy and overall quality of a 3D printed part [13, 3, 44, 66, 33].

Moreover, the complex calibration required to achieve optimal quality on 3D printed objects has influenced many guides such as the SIMPLIFY3D Print Quality Troubleshooting Guide that compiles the most common anomalies in FDM 3D printers, along with images and possible reasons that justify each anomaly [58].

The occurrence of anomalies on printed parts poses a challenge for product quality and overall trust on FDM 3D printers, as printing errors can hinder the intended structure and create frustrating time-consuming troubleshooting procedures to assess the root cause of the anomaly. It is evident that current printing quality assessment methods still heavily rely on extensive human inspection, which is both time-consuming and dreary. In some cases, to assess mechanical strength of the printed structure testing is necessary which can be destructive, resulting in additional material consumption and time to print another piece.

3.2 Print Anomalies

Printing anomalies refer to abnormalities that appear on the printed structure. These anomalies relate directly with printing quality, these can either be visual or extremely detrimental to the printed structure, potentially resulting in destructive outcomes as depicted in Figure 3.1. Anomalies can be categorized as mechanical or induced due to poor parameter calibration during the slicing software stage, Petsiuk and Pearce [56] 2020, proposed a software algorithm that examines layer deposition during the 3D printing process. During the development of the research, the author analyzes three of the most popular Internet hubs of the 3D printing community where common anomalies often appear in the forums threads. During this initial development stage, the authors compiled all relevant errors and created a comprehensive table, Table 3.1, that correlates types of errors with possible reasons. Many present anomalies on 3D prints can result from a combination of problems opposed to a binary reason for the occurrence. This aspect was also evidenced by Brion and Pattinson [7] 2022, where the developed system and dataset considered the correlation of four different parameters.

Mechanical anomalies are challenging to correct with software solutions, broken components require replacement. Induced anomalies often emerge during the slicing stage, some parameters can be adjusted with Geometric Code (G-code) commands that specifically control these elements during ongoing printing jobs. These commands enable fine-tuning parameters which directly impact the final printed structure quality. From this possibility it's evident that printing parameters which have corresponding G-code commands should be considered for future anomaly detection systems leveraging Deep Learning (DL) models. Additionally, parameters such as flow rate percentage, feed rate percentage, printing speed, z offset, and temperatures are all elements that have a corresponding G-code commands. The current value for these parameters can also be retrieved during printing jobs, enabling refined control over the ongoing process and providing additional information for systems to make informed decisions.

Some anomalies are visually similar, this is evidenced in Figure 3.2(b), which shows a lack of extruded material due to poor melting of material, and Figure 3.3, which displays low nozzle temperature anomaly. Both anomalies could be misclassified with one

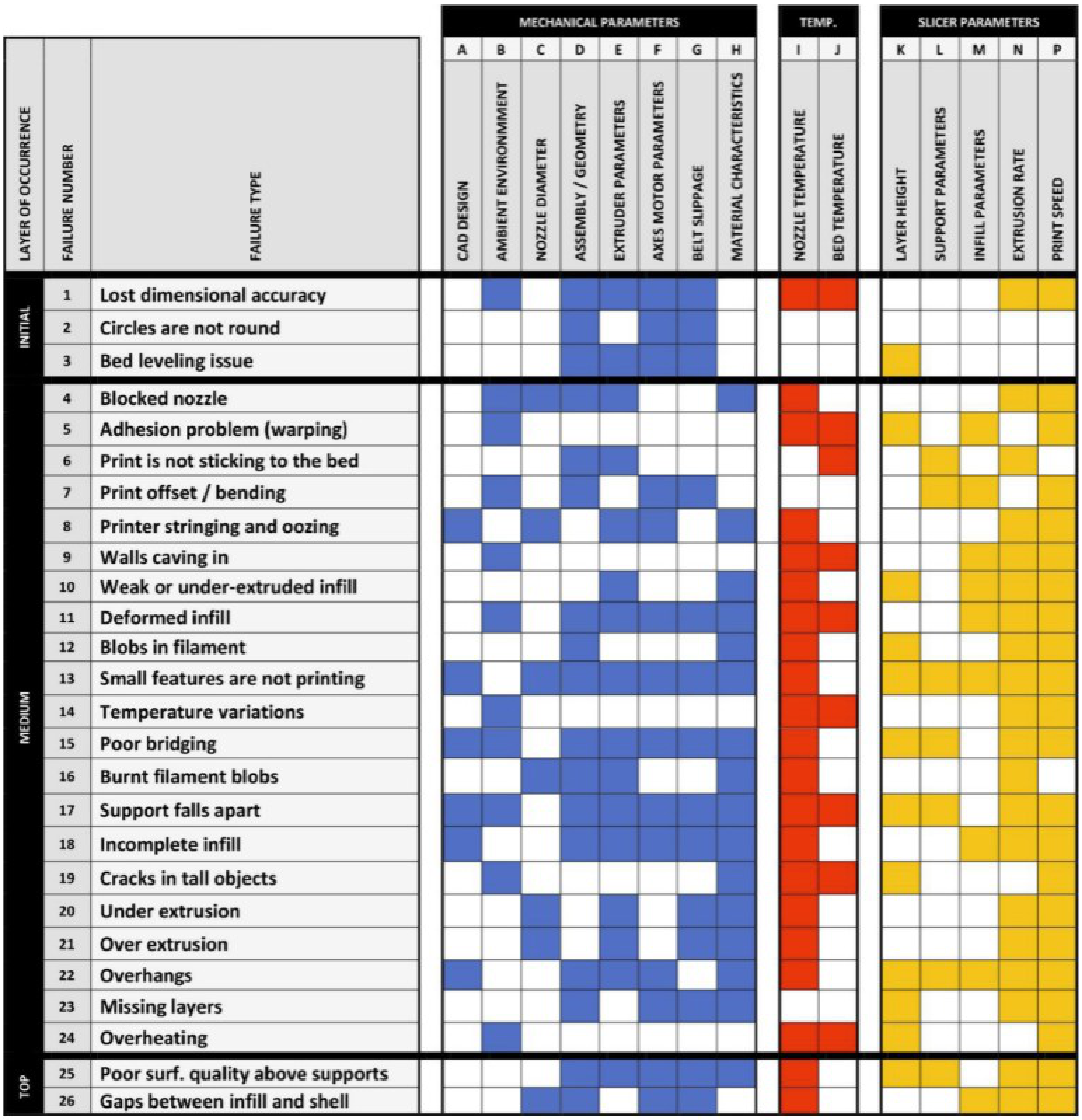


Table 3.1: Correlation between anomalies and possible causes gathered from three different sources as part of the research gathered from Petsiuk and Pearce [56].

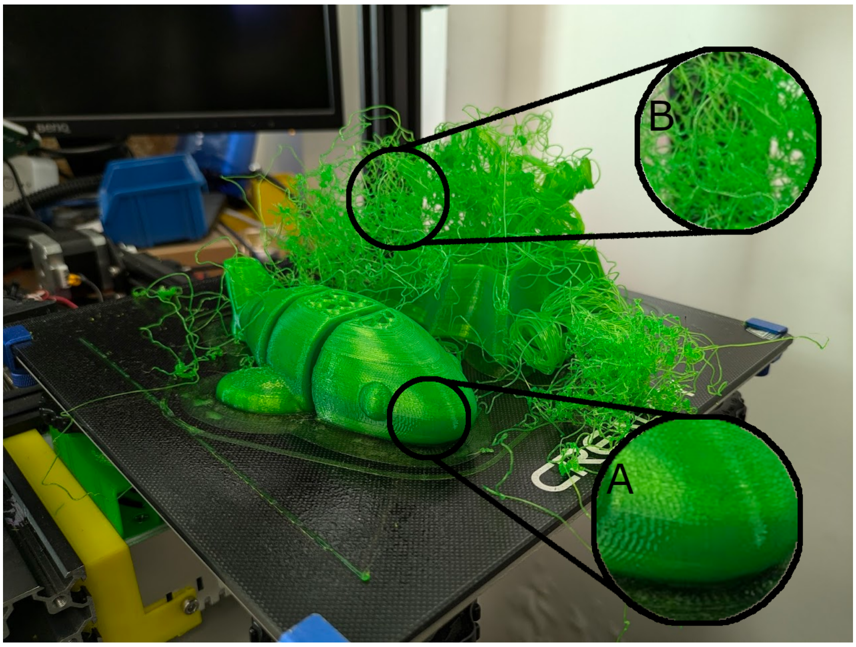


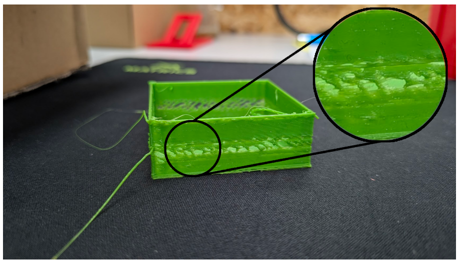
Figure 3.1: An example of failures causing high flow rate (A) and stringing anomaly (B) which resulted in the total loss of the printed objects.

another. Similarly, high extruder temperature can be visually similar as over extrusion anomalies on some instances. The incorrect assessment of the root cause of an anomaly could result in an erroneous correction, such as increasing the extrusion multiplier, while the correct solution however should be increasing the extruder temperature. The outcome would not achieve the desired outcome, and the anomaly should still be present. This type of misdiagnosis highlights the need to consider different parameters for anomaly detection systems in FDM 3D printers. This necessity if further evidenced by the research conducted by Brion and Pattinson [7] (2022), which emphasizes the importance of considering the intrinsic relationships between different settings and anomalies. Consequently, parameters that result in higher rate of erroneous diagnoses should be prioritized.

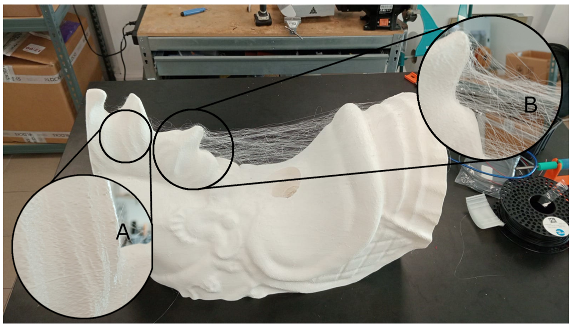
3.3 Filaments

As mentioned in Section 3.2, FDM 3D printers are extremely susceptible to anomalies which are easily transposed onto the printing object affecting the final quality of the printed structure. Filaments also play a significant role in this aspect. The brand and type of polymer used for the filament can impose different printing requirements and the incorrect adjustment of them result in various print anomalies. In Section 2.3, it was disclosed that filaments typically have a diameter of $1.75mm$. Significant variations in these dimensions directly impact the quality of the printed structure, as the extruder momentarily loses control over the filament. Additionally, some filaments require specific ambient temperatures while printing, this requirement can hinder the intended quality as variations of temperature can be mechanically destructive to the printed item with

(a) Over Extrusion anomaly.



(b) Under Extrusion anomaly.



(c) Print exhibiting both Zits/Blobs anomaly (A) and Stringing anomaly (B).

(d) Warping anomaly, the base of the print part is detached from the print bed.

Figure 3.2: Common anomalies that often result from incorrect parameter settings on slicing stage.

anomalies such as poor layer adhesion.

Table 3.2 lists the most common printing filaments, providing an overview of the advantages and disadvantages of each type [65, 47, 53, 71, 37]:

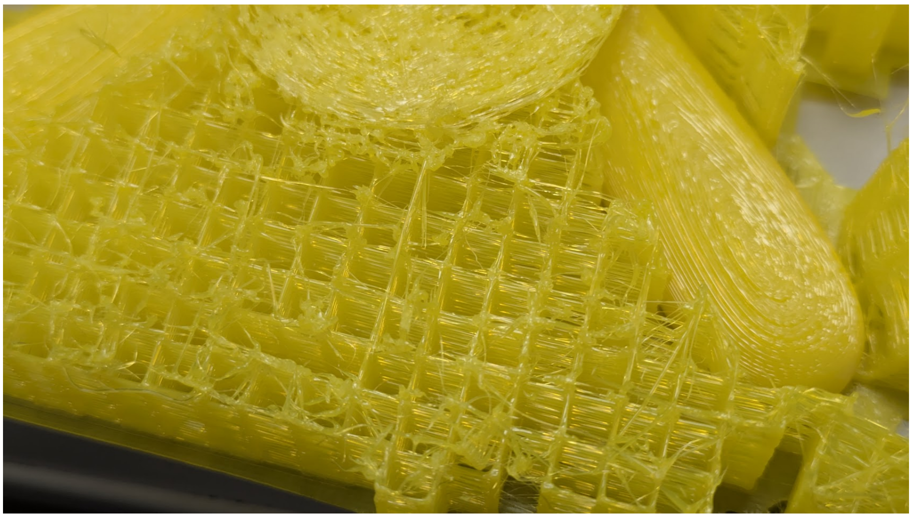


Figure 3.3: Low temperature anomaly, characterized by incomplete extrusion due to insufficient melting of the material.

Table 3.2: Popular filament types and their recommended use properties.

Filament	Extruder Temp.($^{\circ}C$)	Bed Temp.($^{\circ}C$)	Advantages	Disadvantages
PLA	210 - 230	0 - 65	Easy to print, perfect for prototyping; Low required temperature; Can be printed with high velocity;	Less resistant to temperature, printed parts made out of PLA tend to melt if left in the sun; Low Resistance to applied forces;
PETG	230 - 250	60 - 80	Low physical resistance to external forces; Low flexibility; Resistant to temperature; Resistant to UV;	Difficult to print due to stringing; Hygroscopic material, prone to absorb moisture when not in use;
ABS	240 - 260	90 - 110	Very resistant to external temperatures; Very resistant to external forces; Resistant to UV;	Requires a printer enclosure, the ambient temperature must stay high; Prone to printing failures due to warping; Poor layer adhesion;
ASA	240 - 255	80 - 100	Resistant to external forces; Resistant to temperature; Resistant to UV;	Prone to warping; Requires printer enclosure;
NYLON (PA)	250 - 290	70 - 110	Extremely resistant to external forces; Resistant to UV; Extremely resistant to temperature;	Very prone to printing failures; Requires high extruder temperatures, which needs specialized components to handle this; Needs to be printed slower;

3.4 3D Printer Setup

Creating a dataset featuring images of filament deposition 3D printer is crucial. Consequently, the budget-friendly and widely popular Creality Ender3 3D printer was selected as the foundation to advance this objective. The selection process for the 3D printer was carried out to ensure that the current work could be reproducible by the majority of individuals. The Ender3 is simple Cartesian kinematic machine, or *bed-slinger*, where the movement of each axis is controlled by a single stepper motor.

The original Ender3 printer suffers from reliability issues. To increase the printing quality and streamline maintenance in the event of a malfunction, the printer needed to be modified. Considering this concern, all stepper motors were replaced with new higher torque Nema 17 stepper motors. The choice to replace the stepper motors originated from recurrent layer-shift anomalies, common with this particular printer. The replacement stepper motors not only mitigated this type of anomaly, but also enhanced the printers movement control. Additionally, a linear rail was installed on the x axis to replace the worn out V-wheels that were originally installed.

An essential requirement for the printer is the ability to capture the filament deposition images through a nozzle camera, which is mounted on the printers toolhead, directly pointing to the nozzle. Most printers do not come with a nozzle camera, as is the case of the Ender3, so it was necessary to retrofit a camera into the printers toolhead. *VORON Design* [74] is a open-source community project which started on 2015 with the intent of creating and affordable Do it yourself (DIY) 3D printer. Currently Voron Design continues to grow and hosts a repository featuring a range of 3D printer designs with detailed manuals. Apart from open-source 3D printers, Voron also offers a printable toolhead which has gained widespread popularity withing the 3D printing community. The *Voron Stealthburner* is a popularized toolhead from the Voron project. It features a direct drive extruder and is backed by strong community support, offering multiple iterations of the original printable files. Notably, this selection of derivative designs includes the integration of a nozzle camera into the original design [55]. To speed up the nozzle camera integration and enhance the printers precision the retrofitted Voron Stealhburner was printed and assembled onto the Ender3, Figure 3.4. The new toolhead, not only streamlines maintenance in case of hardware failures but also enhances printing quality due to the increased control.

Mainstream open-source firmware solutions for controlling 3D printers with G-code commands include *Marlin* firmware, which operates on a microcontroller board such as the RAMPS 1.4 [82]. Marlin is a precise firmware that produces quality prints, this solution also requires low investment cost, as the printer can be controlled only be the microcontroller board.

The nature of this research requires a hight degree of integration between the different components that compose the work, including the 3D printer control, camera and

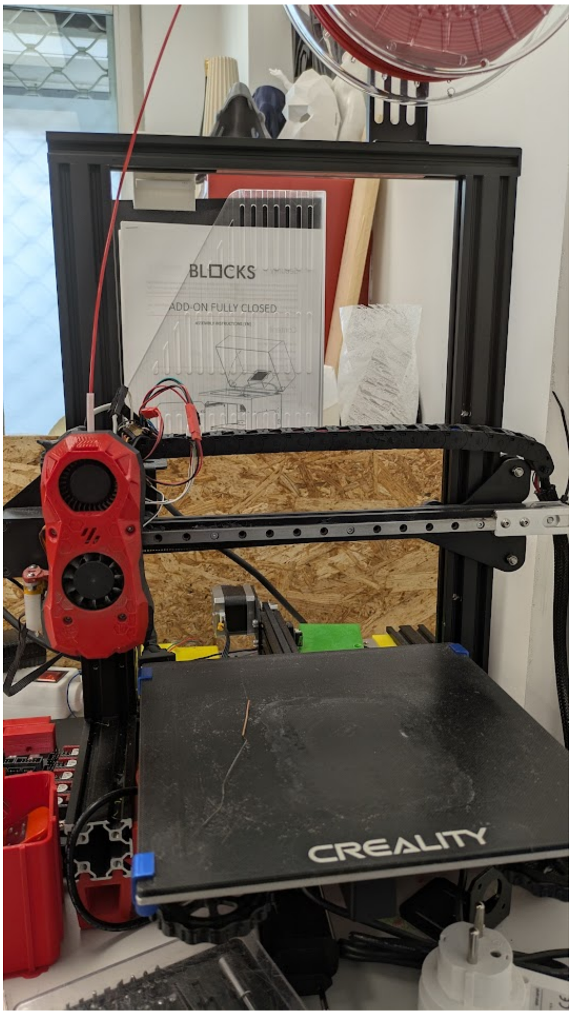


Figure 3.4: Modified Creality Ender3 3D printer with Voron Stealburner and linear rails on the x axis.

deep learning model used to predict anomalies. Marlin does not offer this extend of integration. An additional open-source solution is Klipper firmware, which is becoming increasingly popular, receiving widespread acceptance in the community for its ability for fast printing. It leverages general purpose computers, such as the raspberry pi to handle gcode processing and utilizes the printer's controller board as the middle man for delivering instructions to components including the stepper motors. This configuration transfers the processing demands from microcontroller to more computationally powerful computer, this architecture has hindered Marlin firmware to reach the same printing speeds as Klipper [76, 21, 82]. The ability to run the printers firmware on a general purpose computer is a great advantage as it permits using more advanced features such as a more intuitive GUI and network connectivity, additional software packages can be used, including software that enables the control of the camera.

For the previously highlighted reasons, to manage all components within a single package, Klipper firmware was chosen for the 3D printer, running on a ODROID C4 Single Board Computer (SBC), equipped with an Armlogic S905X3 CPU and 4 GiB of

DDR4 RAM. Debian Bookworm version 12.0, was used as the operating system for Klipper firmware to run. For the printers controller board, the SBC was paired with a SKR mini E3 v1.0 from BIGTREETECH, both of which was subsequently installed onto the modified Ender3 printer. Initially, the chosen camera was a 3DO Nozzle Camera, capable of capturing images in 4K resolution, but as the work progressed, other off-brand cameras were also utilized that only permitted lower resolution image capturing.

3.5 Computer Vision Model

From the literature disclosed in Section 2.5, it is evident that the advantages provided by computer vision models, which capitalize deep convolutional neural networks, result in more capable architectures for tasks involving image classification. The model developed in the research conducted by Brion and Pattinson [7] (2022), demonstrated state-of-the-art results, while considering intrinsic relationships between anomalies and possible root causes, as highlighted as a critical aspect for anomaly detection in Section 2.5. For these reasons, the Collaborative Autonomous Extrusion Network (CAXTON) model was selected as the backbone model for the system developed in this research.

In the first subsection of this section is described the CAXTON model, along with its backbone, which was modified in this research, with the introduction of an additional layer (Squeeze and Excitation Layer (SELayer)). Subsection 3.5.2 presents the SELayer, whereas subsection 3.5.3 discloses the proposed network with three different configurations.

3.5.1 CAXTON Network

Brion and Pattinson [7] introduced a multi-head deep neural network CAXTON, as disclosed in Section 2.5, capable of classifying four parameters - flow rate, feed rate, z offset and hotend temperature - with the labels ”good” ”low” of ”high” by training the model on material deposition images gathered from on going print jobs.

CAXTON backbone consisted of using the **Attention-56** Residual Attention Network (RAN), introduced by Wang et al. [75] (2017), on the paper ”Residual Attention Network for Image Classification”. This RAN model integrates a combination of residual connections and stacked attention mechanisms into existing state-of-the-art deep residual learning frameworks, to improve the networks feature representation abilities and performance. The network is based on three properties [75]:

- **Stacked structure**: Multiple attention modules are stacked, these modules consist of a trunk branch and a mask branch, as depicted on Figure 3.7. The stacked configuration effectively enhances the networks ability to capture important features maps from the input images. By leveraging a mixed attention mechanism each module captures different attention types such as spatial and channel attention, adaptively learning specific attention aware features as the network deepens. This property also enables the network to incrementally refine learned attention masks. This is achieved without increasing computational complexity and allows the network to scale up to hundred of layers.

- **Attention Residual Learning**: Due to problems that emerge from stacking multiple attention modules, the network utilizes soft attention masks in every mask branch present in attention modules to avoid performance drops. This technique increases the network abilities to extract important features and suppresses less relevant ones by applying the generated attention masks. Furthermore, the use of residual connections ensures a consistent gradient flow during backpropagation, which helps alleviate issues like vanishing gradients. This also enhances the network's robustness against noisy labels by serving as gradient filters throughout the backpropagation process as the mask branch inhibits wrong gradients to update the trunk branch. Considering $M(x)$ as the output from the masked branch ranging from $[0, 1]$, and $F(x)$ as the generated features from deep convolutional networks, the output from an attention module, $H(x)$, is represented as:

$$H_{i,c}(x) = (1 + M_{i,c}(x)) * F_{i,c}(x) \tag{3.1}$$

- **Bottom-up top-down structure**: This property is a key component of attention modules, it captures global and local contexts to generate attention masks by adding soft weights on features. The global context, bottom-up pathway, allows the module to capture global context information by downsampling the input feature maps, using operations such as max pooling. This increases the receptive field and allows the network to capture broader patters. Local context, top-down pathway, combines the previously captured global context with local information through upsampling the received feature maps back to their initial dimensions by interpolation. This ensures that the produced attention masks not only capture global contexts but additionally local information. This processes is done in a feedforward fashion, meaning that it performs the operations in a single pass through the network, making the mechanism efficient for deep networks.

The basic building block of this network is a residual unit, these units are designed to, much like the attention modules, help suppress the vanishing gradients problem by employing a skip connections, usually bypassing one or more operations. This basic unit is depicted on Figure 3.6.

The three principles highlighted previously are what makes the RAN model so attractive for image classification tasks. These principles not only enhance feature representation, allowing the network to focus on the most relevant parts of the image and thereby increase the model's accuracy, but also leverage the attention mechanism ability to suppress noisy labels during the models training, which is another key feature of the network. CAXTON leverages **Attention-56** network which stacks three attention module stages:

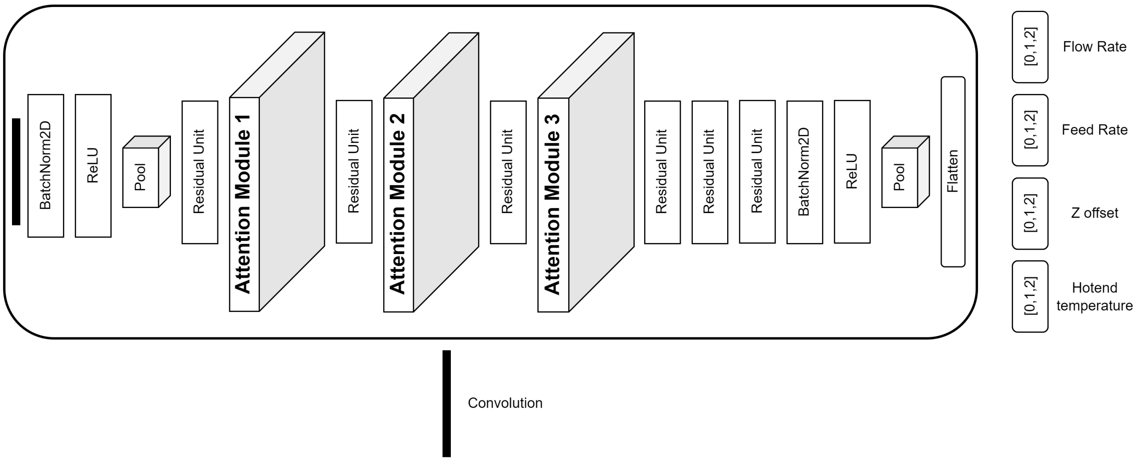


Figure 3.5: Collaborative Autonomous Extrusion Network (CAXTON) architecture (adapted from [7]).

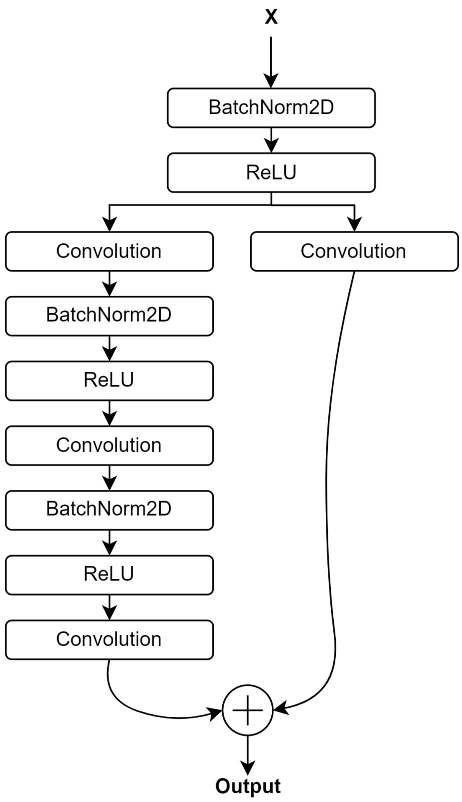


Figure 3.6: Residual Unit architecture.

1. **Attention Module Stage 1** - With input size of 56x56, this stage captures a broader amount of patterns from the input and helps suppress background information, Between the bottom-up and top-down feedforward structure it features $k = 2$ amount of residual units.

2. **Attention Module Stage 2** - Input size of 28x28, refines the attention aware feature

maps produced from the previous stage by supplementing detailed local information. Features $k = 1$ residual units on the mask branch.

3. **Attention Module Stage 3** - Input size of 14x14. On this stage the network comprehends the input image, combines the features extracted on the previous layers and refines them one last time for the network to only focus on the most relevant features maps. Between the bottom-up top-down feedforward structure there are no residual units, $k = 0$.

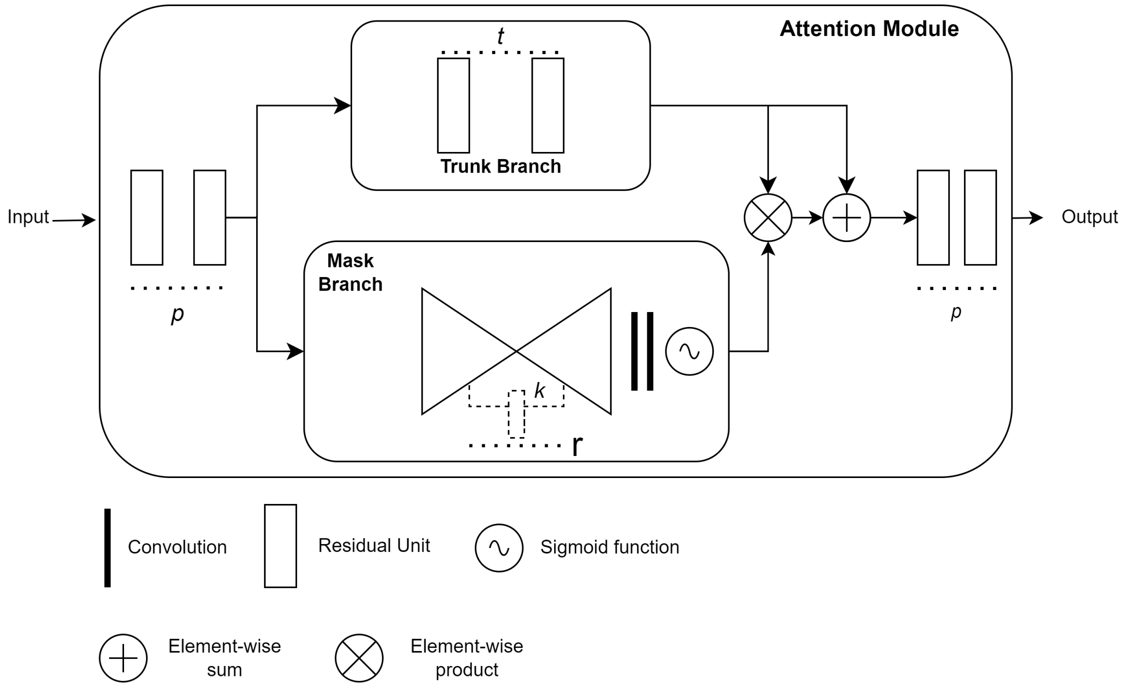


Figure 3.7: Architecture of an attention module (adapted from [75]).

The hyper-parameter r, depicted on Figure 3.7, is a configurable amount of residual units between the pooling layers on the mask branch, p corresponds to the number of residual units before splitting the input into the trunk and mask branch, t is the number of residual units on the trunk branch, and k is the amount of residual units between the bottom-up and top-down structures of the mask branch, this last parameter is specific for each attention modules stage. On the CAXTON backbone, Attention-56 RAN, the hyper-parameter are set to $r = 1, p = 1, t = 2$ [75].

Attention modules become even more important when considering images that contain a substantial amount of information as is the case of filament deposition images, Figure 3.8, the robustness to noise and the ability for the model to focus on important regions of the image becomes even more critical aspect for the development of the current model.

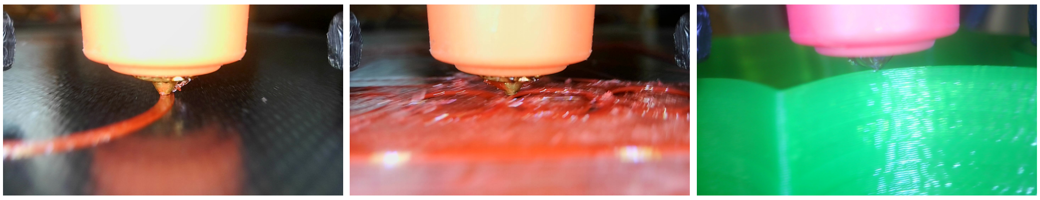


Figure 3.8: Examples of a filament deposition images with 1920x1080 pixel dimensions, captured during a print job.

An important architectural aspect of the CAXTON model is its ability to label four different classes, this is achieved by creating a multi-head setup on the output of the networks backbone. By integrating onto the end of the network four different linear transformations, represented in Equation 3.2, that receive the output of the backbone as input. Each linear transformation is assigned to predict a distinct task. In this instance, it involves labeling as either ”low” (Value 0), ”good” (value 1) or ”high” (value 2) on each selected parameter class: ”flow rate,” ”feed rate,” ”z offset,” and ”hotend temperature”.

$$y = x * A^T + b \tag{3.2}$$

where x is the input, A is the produced weight matrix and b is the bias vector.

3.5.2 Squeeze and Excitation Layer

Hu et al. [27] (2020), on the paper ”Squeeze-and-Excitation Networks”, in This study introduces the SELayer, a novel component that can be seamlessly incorporated into existing deep architectures to adaptively recalibrate features by leveraging both spatial and channel-wise information. This innovat ive architectural unit not only improves the neural network’s ability to represent features but does so with a relatively lightweight mechanism that only slightly increases the networks complexity. This state-of-the-art block has demonstrate great results, even winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2017 competition.

As the name suggest, this is achieved by the following operations:

1. **Squeeze**: Compresses spatial dimensions of the input feature map. This operation consists of applying global adaptive pooling to each channel of the feature maps, compressing the input channels $\mathbb{R}^{H*W*C}$, to a single numeric value $\mathbb{R}^{1*1*C}$.

2. **Excitation**: This operation accepts the compressed channel from the previous step and fully captures channel-wise dependencies providing the layer to adaptively recalibrate. This operation generates weights for each channel and adaptively recalibrates the channel descriptors. This is achieved through a two fully connected layers, with a Rectified Linear Unit Function (ReLU) activation function in between, followed by a sigmoid activation function at the end. The first fully connected layer reduces the dimensionality of the squeezed vector $\mathbb{R}^{H*W*\frac{H}{r}}$, r is the **reduction ratio**

factor, the second fully connected layer restores the vector to its original dimensions. The final activation function, sigmoid function, ensures that the produced weights are in the range $[0, 1]$.

After squeeze and excitation operations the output is scaled back to the original size. In contrast to other attention mechanisms, squeeze-and-excitation layers employ **channel-wise** attention, which allows the layer to capture the interdependencies between channels and recalibrate feature maps, allowing the network to focus on the channels that contain the best information.

3.5.3 Proposed Network

As discussed in subsection 3.5.1, the CAXTON multi-head network utilizes RAN Attention-56 model as its core. This network demonstrates an incredible ability to enhance and extract important features from input images, a benefit derived from the stacked attention mechanism architecture. The significance of attention mechanisms is evident, which plays a pivotal role on enhancing the networks capabilities to feature representation, which enables the network to concentrate on pertinent information and reduce noisy gradients (thus addressing the vanishing gradients problem), ultimately increasing the models performance.

To enhance the model's capability to focus on significant features and suppress the less relevant ones, the present study suggests incorporating an SELayer onto the CAXTON backbone, in three different configurations.

This recommendation is based on the SELayer's capacity to capture interdependencies between channels, which is enabled by the **channel-wise** attention mechanism present on the excitation phase of the layer, as explained on subsection 3.5.2. It is believed that the addition of SELayer will increase the models performance without significantly increasing the models complexity.

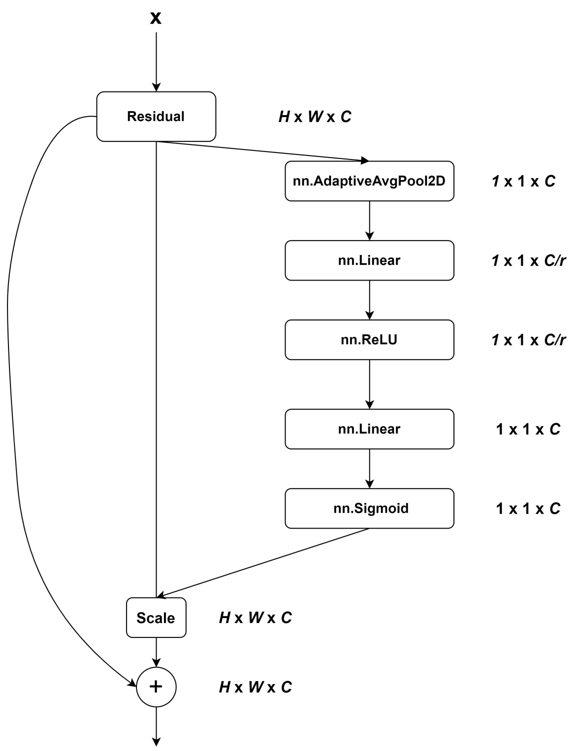


Figure 3.9: Squeeze and Excitation used with Residual Blocks modules (adapted from [27]).

The initial configuration adheres to the suggested positioning of the SELayer as recommended by the layer's creators. In this case it, the layer was added after each residual unit that follows an Attention Module as depicted on Figure 3.10.

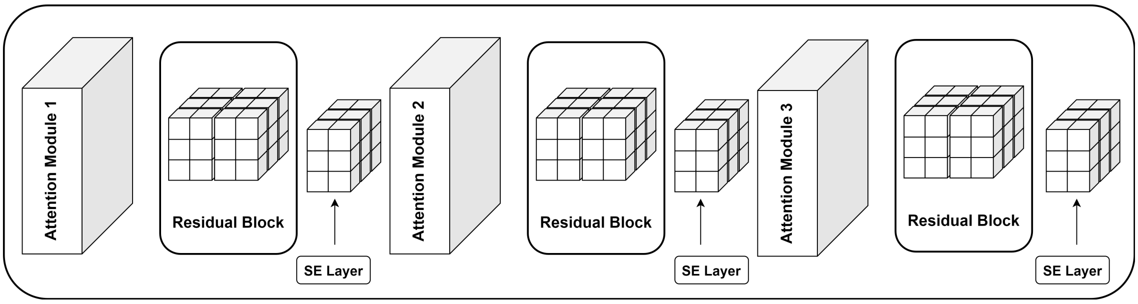


Figure 3.10: Squeeze and Excitation layer after each Residual Blocks on attention modules.

The other following configurations involve adding a SELayer onto residual units, Figure 3.11, the placement of these modified units is what formulates the last two configurations.

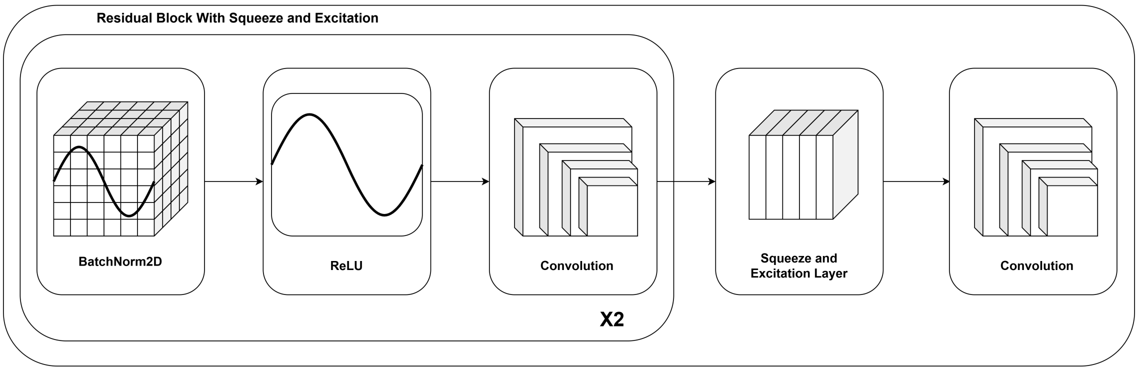


Figure 3.11: Squeeze and Excitation Layer relative position on Residual Block architecture.

The second proposed configuration is achieved by introducing the modified residual unit enhanced with a squeeze and excitation layer, were utilized to . The first model involves substituting the residual blocks following the attention modules with the modified residual units. This configuration is depicted in Figure 3.12.

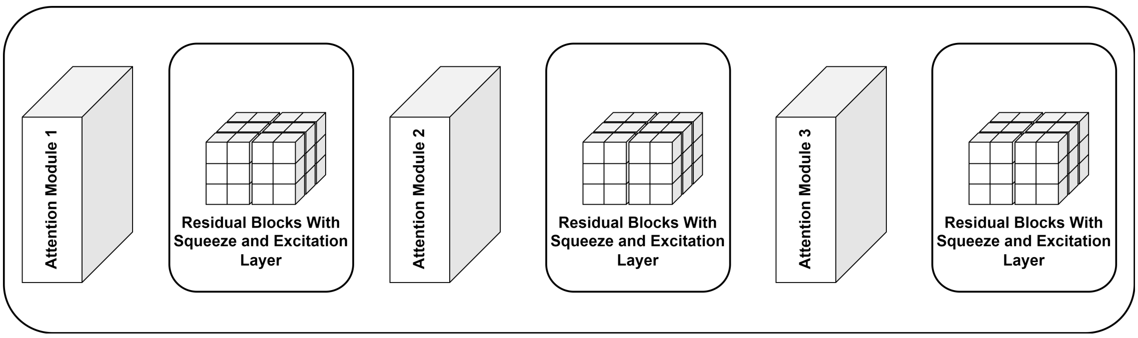


Figure 3.12: Representation of the modified residual units with squeeze and excitation, after Attention modules.

The last, and final proposed model, is designed by adding the SELayer modified residual unit, onto the attention modules, Figure 3.7. In this configuration, every residual unit, is replaced by its modified counterpart.

The Squeeze and Excitation layer was designed following the guidelines of the paper, with the channel reduction ratio (r) set to 16. This value is considered to be the optimal balance between complexity and enhanced performance, as suggested by the authors.

3.6 Dataset

The development of a DL computer vision model entails the need of extensive amounts of data. The CAXTON dataset from Brion and Pattinson [7] (2022), contains 1,272,273 Joint Photographic Experts Group image file format (.jpg) nozzle images, captured during printing jobs. This data collection gathers images from 190 distinct prints, each featuring various geometries and filament colors. This diversity is crucial for the construction of a robust dataset, it ensures that the model has access to a broader range of data to train from.

The structure of the dataset is composed of a hierarchical directory separated in folders named with the prefix *print_* and the suffix consisting of a number from 0 to 189 indicating the print number. Each print folder contains images from the current print with four additional Comma-separated values (CSV) files. The naming convention for these files is *image-x*, where *x* is the index for the image starting from 0. Additionally, the *.csv* files, which contain directories and printer information corresponding to the time that the image was captured, are stored within each print folder.

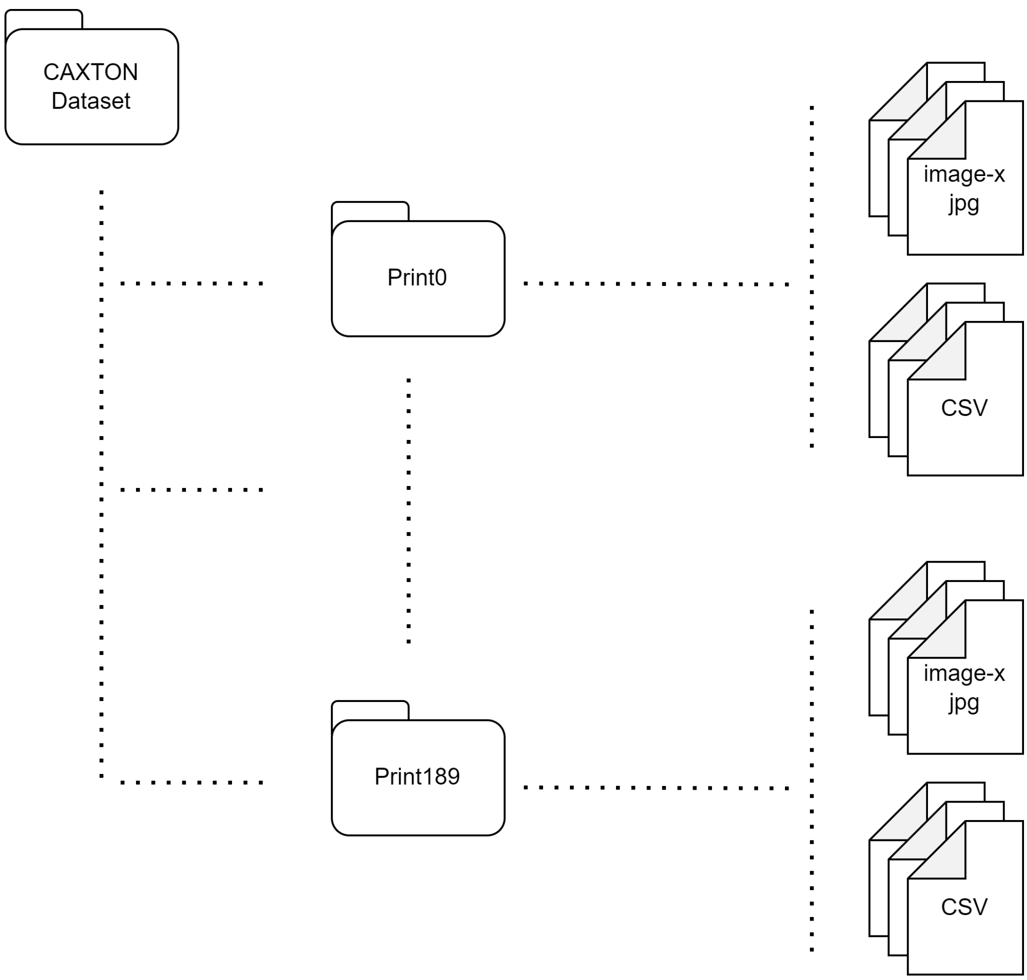


Figure 3.13: Diagram of the folder structure from the CAXTON dataset.

Information stored within CSV files relate to the captured images and corresponding printer parameters. Each entry row contains an image path that relates to the current print folder and values for flow rate, feed rate, z offset and extruder speed, the keywords for each columns are as follows:

`img_path` - Contains the relative image path of the images, in relation to the software.

`timestamp` - Timestamp of the image taken.

`flow_rate` - The value for the current printer flow rate.

`feed_rate` - The value for the current printer feed rate.

`z_offset` - The z offset value for the relation nozzle tip / bed.

`target_hotend` - The temperature value the printer hotend is supposed to be on.

`hotend` - The actual hotend temperature registered.

`bed` - The actual bed temperature registered.

`nozzle_tip_x` - Position of the nozzle in x axis.

`nozzle_tip_y` - Position of the nozzle in y axis.

`print_id` - Print identification.

`flow_rate_class` - The flow rate class. Assuming values of either $0, 1, 2$.

`feed_rate_class` - The feed rate class. Assuming values of either $0, 1, 2$.

`z_offset_class` - The z offset class. Assuming values of either $0, 1, 2$.

`hotend_class` - The hotend class. Assuming values of either $0, 1, 2$.

The dataset developed in this work is designed to test the repeatability for new FDM 3D printer mechanical configurations and serve as an extension for the CAXTON dataset developed on the research conducted by Brion and Pattinson [7] (2022). The dataset on the current study, similarly adheres to the same structural framework established in the prior work. This includes no only the storage of images captured from the nozzle within and identical file directory hierarchy, as in Figure 3.13, but also gathers information following the CAXTON dataset's labels and keywords, which include the items previously highlighted.

To accommodate the increasingly changing FDM 3D printing landscape, the developed dataset utilizes the **Klipper firmware's** capability to report various parameters of the current print job, to add more print information onto the CSV files. New information is added into the dataset while ensuring backwards compatibility with the original dataset, this is achieved by preserving the original keywords listed above and only adding new information to the CSV files with new keywords as outlined in the list below:

`bed_temperature` - The current temperature of the bed plate.

`target_bed` - The target temperature for the printers bed.

`speed_factor` - Klippers speed multiplier in percentage.

`speed` - The current print speed in mm/s.

`toolhead_pos` - An array that contains the absolute position of all toolhead steppers, including the extruder in mm.

`gcode_move_pos` - An array that contains the relative toolhead position as reported by the gcode in mm.

`motion_report_live_pos` - Absolute position of the printer stepper motors.

`estimated_print_time` - The estimated print time calculated from Klipper.

Klipper firmware structure permits the addition of extra modules that operate in synchrony with the rest of the system. Higher level code is mostly written using Python programming language, the directory hierarchy, represented on Figure 3.14, consists of three main sub-directories: the *kinematics* package, which is a set of python modules that contain class definitions for the movement systems, some of which were described in section 2.3; the *chelper*, another package containing python bindings, these allow function calls defined in C programming language, vital for movement control defined on the kinematics package; and an additional sub-directory, named *extras*, this contains a list of module classes written in Python that confer most of the functionalities available in Kipper, among the many modules some include class definitions that enable bed leveling, G-code commands, sensors definitions, displays, fans, and thermistors. Klipper knows which modules to load during runtime by using an additional module that gets and reads Configuration file format (.cfg). For a printer to work with the Klipper firmware, a *.cfg* file with the name *printer.cfg* must be specified, this file contains technical information about the printer and additional definitions to load correspondent modules.

Mainsail, a user interface designed specifically for the Klipper system, which software enables the control of Klipper system through a web-browser, was utilized for the development of this dataset. The User Interface (UI)l allows operators to initialize prints, receive information about the printer, and even change configuration files that Klipper uses [77]. Moreover, the software enables camera monitoring if one is configured on the 3D printer, as in the case of the current system. To this end, mainsail crew provides another software package, **Crowsnest**, this contains additional wrapper scripts that facilitate camera streaming by using leveraging **uStreamer**, a lightweight server that streams Motion-JPEG (MJPEG) video. To improve the printer's operation and control in this project, the Mainsail user interface and Mainsail Crowsnest were utilized.

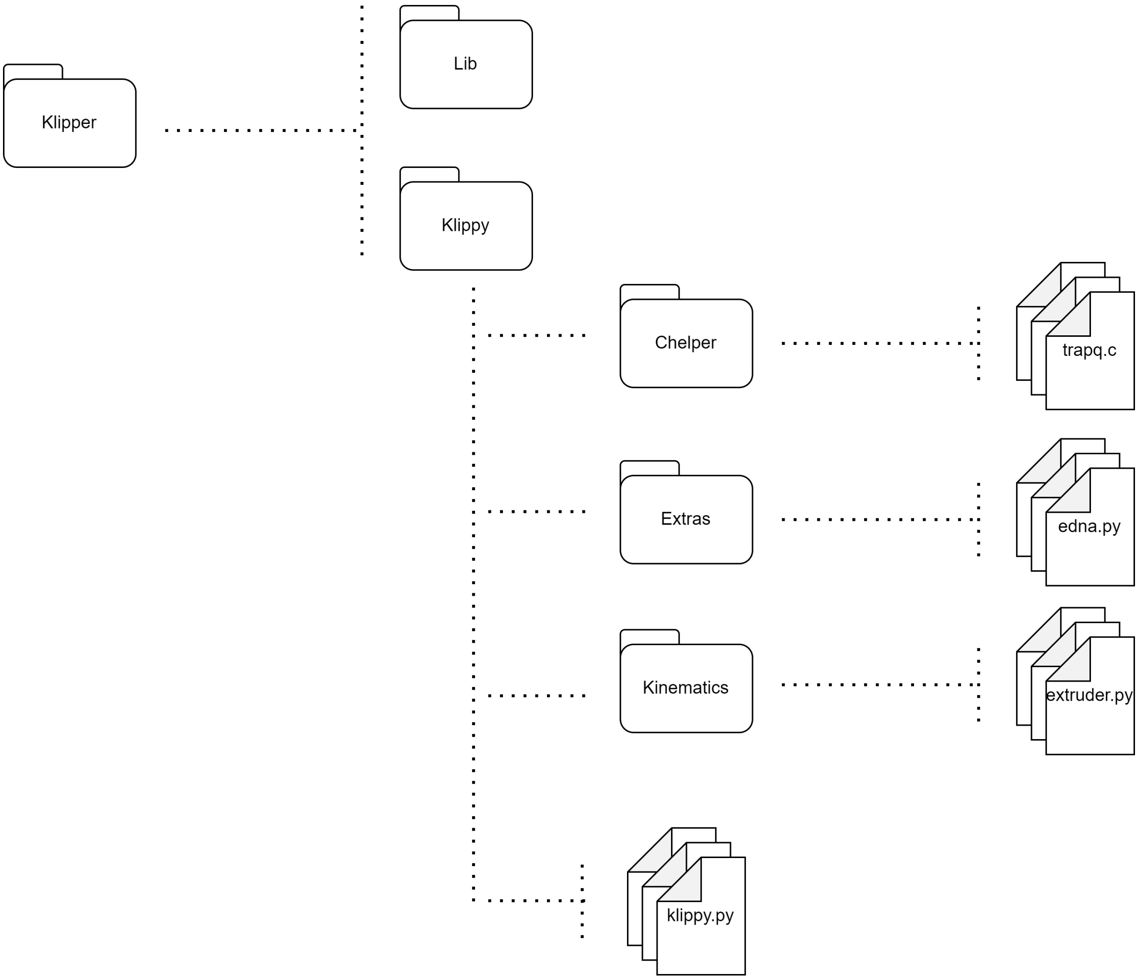
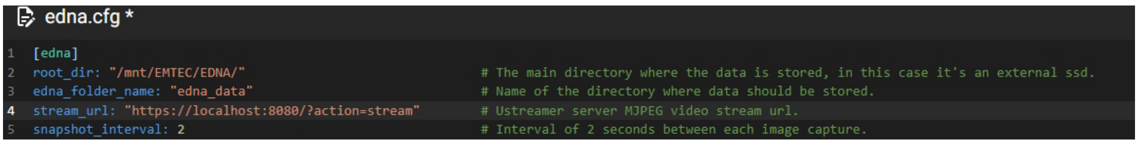


Figure 3.14: Klipper firmware directories schematic with the positioning of the edna python file module.



```
edna.cfg *
[edna]
root_dir: "/mnt/EMTEC/EDNA/"                          # The main directory where the data is stored, in this case it's an external ssd.
edna_folder_name: "edna_data"                         # Name of the directory where data should be stored.
stream_url: "https://localhost:8080/?action=stream"   # Ustreamer server MJPEG video stream url.
snapshot_interval: 2                                  # Interval of 2 seconds between each image capture.
```

Figure 3.15: Edna module configuration file that indicates Klipper Firmware to load the designed edna.py class.

As mentioned above, Klipper uses configuration files to know which modules to load at run-time. A configuration file, Figure 3.15, contains technical information on the developed module and included as a dependency into the main *printer.cfg* file. The defined arguments include module-specific parameters, the root directory for storing the dataset, a designated folder name for the dataset, the server Uniform Resource Locator (URL) for the MJPEG camera stream, which operates on the same device as the Klipper system, and an additional float value specifying the interval for capturing images.

The developed module consists of a Python class containing all necessary methods for

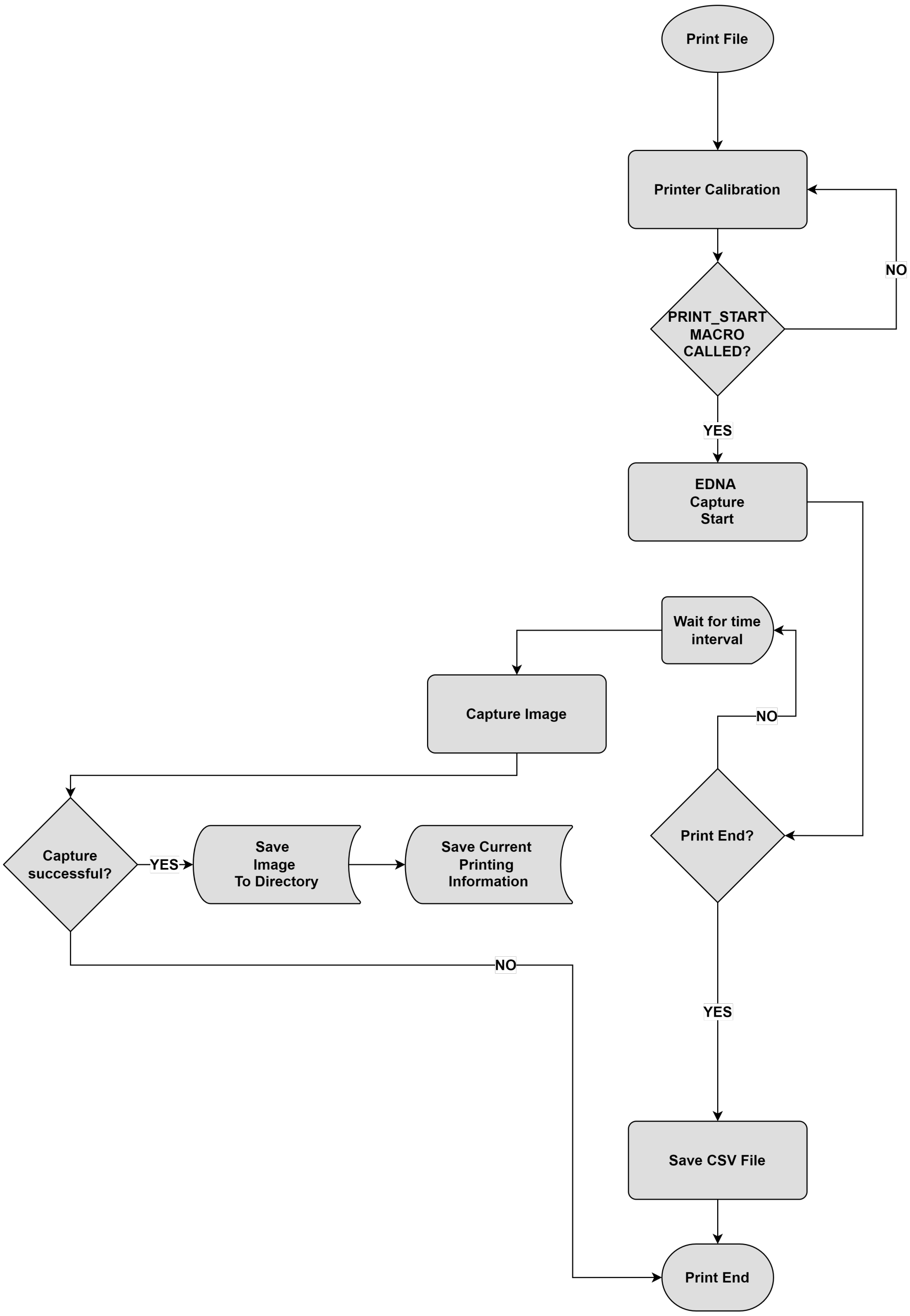


Figure 3.16: Flowchart depicting the image capturing steps.

automatically acquiring and saving images of filament deposition and interacting with the Klipper firmware system. Figure 3.17 contains the class diagram with the methods and internal class variables used during the runtime of the module. Image gathering was achieved using Python Open Computer Vision Library (OpenCV) library package. The method *take_and_save_snapshot* receives the uStreamer server URL defined on the crowsnest configuration file. Using this URL OpenCV *VideoCapture* class can access the video stream and retrieve and save a frame of the video stream .

Figure 3.17: Class diagram for the purposed module class.

For an easy visual reference, Figure 3.19, contains a flowchart that describes the developed automatic image capturing process, this starts when the *PRINT_MACRO* is called when a print file is selected for printing. Macros in Klipper are essentially user defined G-code commands and impose no restrictions on the naming convention. The required macro for the automatic image gathering initialization is a commonly used by the community, the *PRINT_START* macro usually contains user specific initialization processes that are required for the printer to start printing. These user defined procedures usually include instructions for bed leveling calibration and filament sensor activation's. Given the widespread use of this macro, the developed module defines a base PRINT_START macro that users can re-implement if necessary, Klipper enables macro re-implementation as long as the previous macro definition is renamed. The Ender 3 printer used in this work, utilizes the PRINT_START macro for initial calibration and preparation, Figure 3.18 described the functionality enabled by the re-definition and renaming of the base edna PRINT_START macro to PRINT_START_OLD. When the last process is executed, the module automatic image capturing starts its execution.

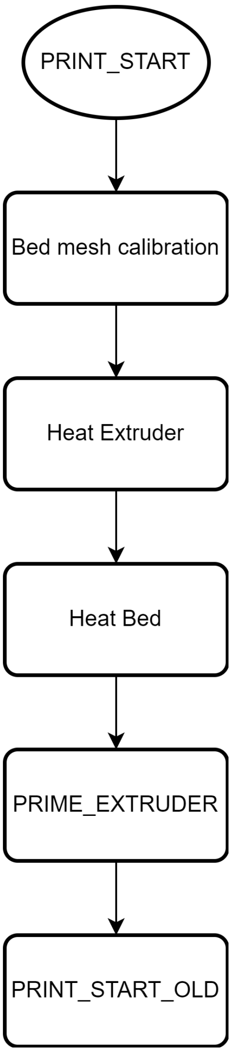


Figure 3.18: Flowchart with the redefined PRINT_START macro process execution before automatic image capturing starts.

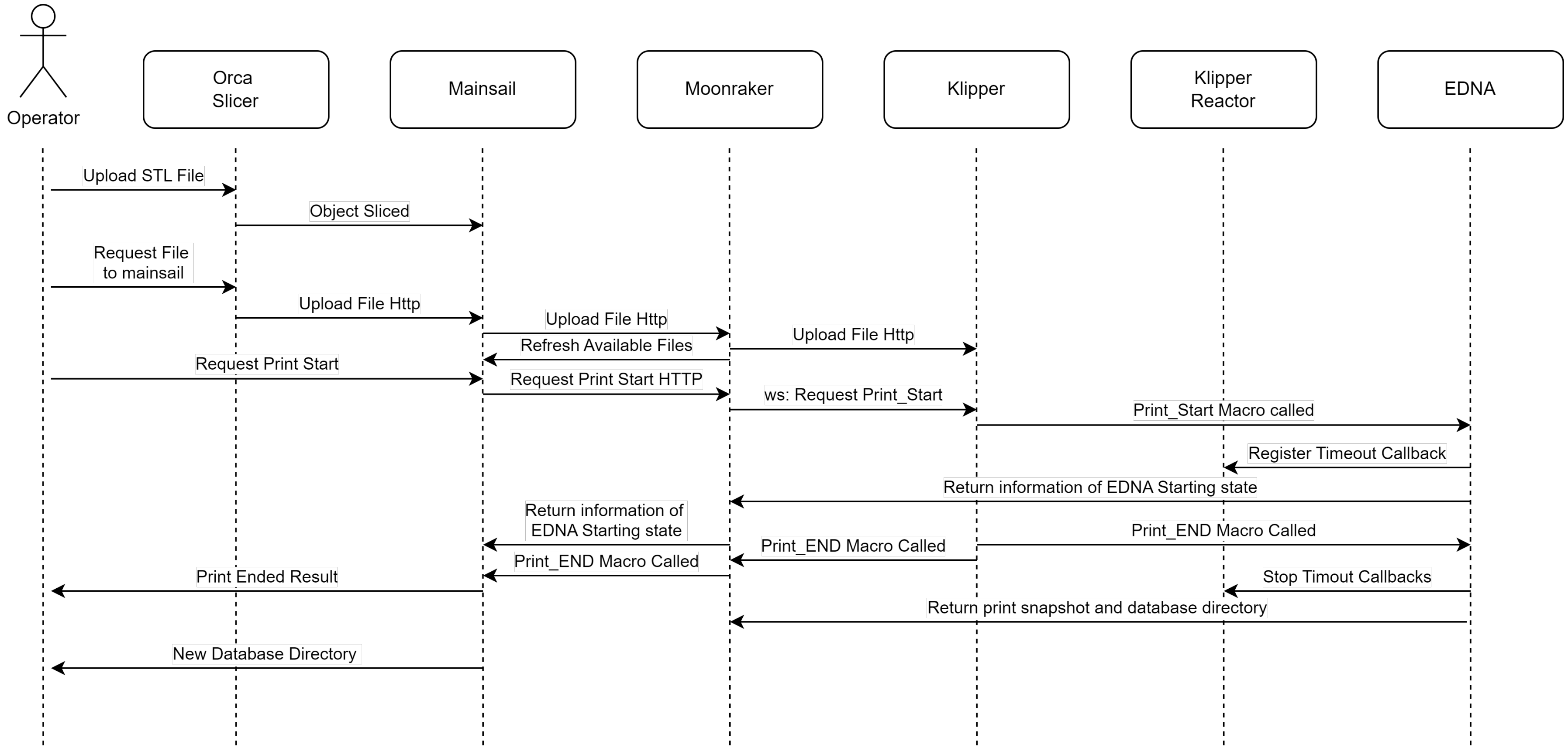
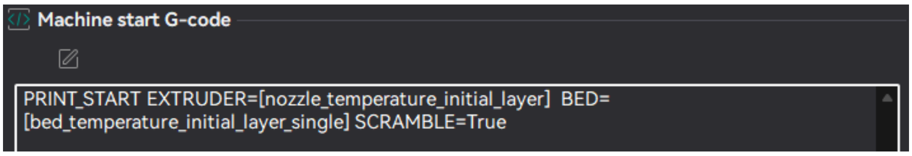
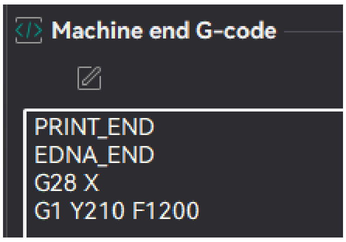


Figure 3.19: Interactions diagram - Depicts the communications sequence starting from the slicing of a three-dimensional object to the start of a print file and EDNA that enables the creation of a dataset with the information of the current print.

Most slicers allow the injection of G-code commands onto the generated *.gcode* file. In order to call the PRINT_START macro, this G-code command was programmed to be injected onto the sliced file before the execution of the models G-code movements. Additionally, to stop the automatic image capturing, the module defines EDNA_END macro which concludes the image gathering and saves the gathered images onto a *.csv* file on the same directory of the images.



(a) G-code command injected before the sliced model G-code commands.



(b) G-code command injected after the sliced model G-code commands finish.

Figure 3.20: Injection of G-code commands during the generation of a *.gcode* file.

The argument SCRAMBLE is a boolean that indicates if the module needs to attribute automatic class definitions and select parameters for the selected gathered parameters. This functionality is extremely important for the creation of the dataset, the argument gives an indication to the model if there is a need to ”create problems” on the print. Although it results in print failures, its purpose is still relevant as it enables creation a diversified dataset where various problems occur at the same time due to different reasons. When the argument is set to True, a number of layers, inferior to the printing object total number of layers, is selected. As in the case of capturing images this is achieved with a timeout function, the module knows when to change parameters when 10% of the total amount of layers is reached. This relation was selected to allow the module to alter parameter values and capture images featuring various combinations of the labels ”low,” ”good,” and ”high” for the designated classes.

To automatically select which label to attribute each class, ”low”=0, ”good”=1 and ”high”=2. The scramble option automatically selects which numeric label to attribute for each selected class, ”feed rate”, ”flow rate”, ”z offset” and ”hotend temperature”. The range differs for each class, but follows the exact same rules, during the print, if the captured parameter value falls bellow the specified range, which starts at the initial print value, then the value will be 0 representing ”low”, if it is within the range then its 1 as in ”good”, and finally if it is above the range then 2 for ”high”. This automatic labeling can

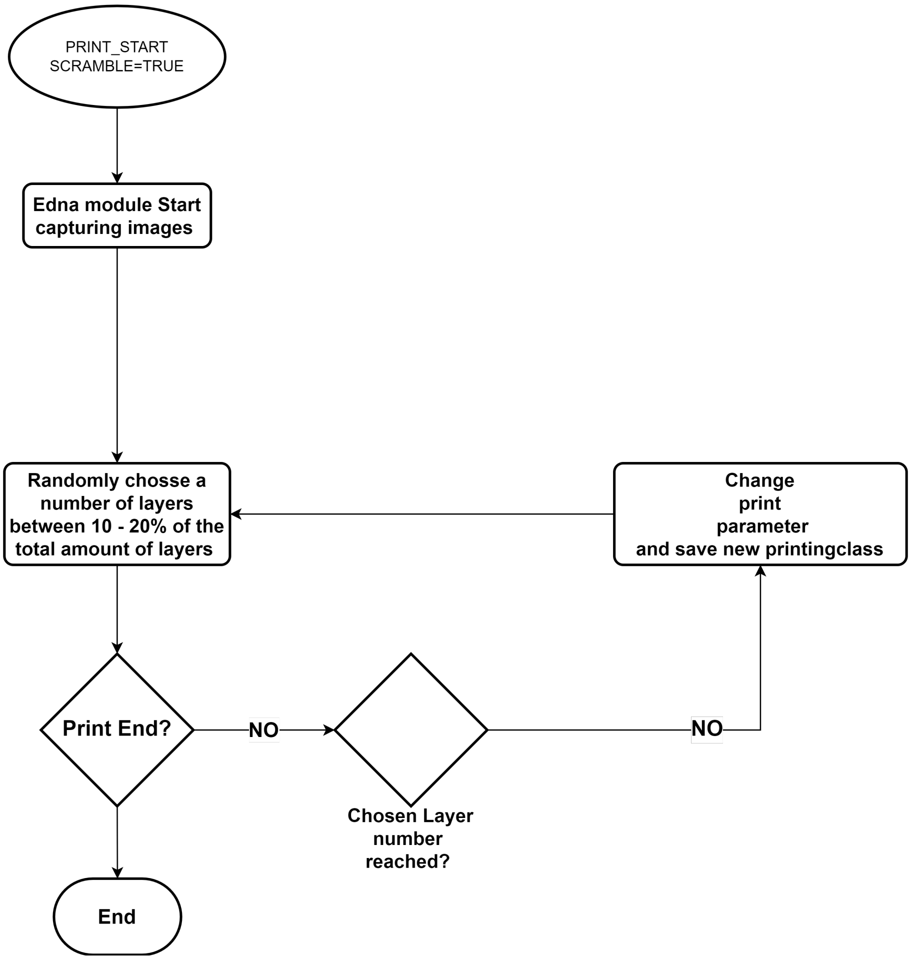


Figure 3.21: Flowchart for the scrambling option that enables automatic class attribution when enabled.

be seen on the following equations:

$$FlowRate(value) = \begin{cases} 0 & \text{if } value < k - 20 \\ 1 & \text{if } k - 20 \geq value \leq k + 20 \\ 2 & \text{if } value > k + 20 \end{cases} \tag{3.3}$$

Where k is the initial value of the parameter, $k = 100$.

$$FeedRate(value) = \begin{cases} 0 & \text{if } value < k - 40 \\ 1 & \text{if } k - 40 \geq value \leq k + 40 \\ 2 & \text{if } value > k + 40 \end{cases} \tag{3.4}$$

Where k is the initial value of the parameter, $k = 100$.

$$zoffset(value) = \begin{cases} 0 & \text{if } value < k - 0.010 \\ 1 & \text{if } k - 0.005 \geq value \leq k + 0.005 \\ 2 & \text{if } value > k + 0.010 \end{cases} \tag{3.5}$$

Where k is the initial value of the parameter.

$$hotendtemperature(value) = \begin{cases} 0 & \text{if } value < k - 10 \\ 1 & \text{if } k - 10 \geq value \leq k + 10 \\ 2 & \text{if } value > k + 10 \end{cases} \tag{3.6}$$

Where k is the initial value of the parameter.

$$Bedtemperature(value) = \begin{cases} 0 & \text{if } value < k - 10 \\ 1 & \text{if } k - 10 \geq value \leq k + 10 \\ 2 & \text{if } value > k + 10 \end{cases} \tag{3.7}$$

Where k is the initial value of the parameter.

For the actual printing phase of the dataset, a variety of prints that demonstrated complexity were downloaded from popular repositories. This choice was made to ensure that the dataset includes a wide range of images, under various conditions and geometries, thereby guaranteeing a diverse amount of examples within the dataset. Moreover, to create a robust dataset, much like the original CAXTON dataset, three different filament colors where chosen: Ruby, Orinoco and Chlorophyll - as described from the manufacturer - these account for extensive range of colors that are commercially available for purchase.

3.7 Monitoring Prototype

All code created during the development of the present work was done using Python version 3.10.2. The computer vision model was also written in Python with the machine learning library PyTorch version 2.4.1+cu121 and to reduce boilerplate code lightning version 2.4.0 was utilized. Additionally, all models were trained on a Laptop running an Intel Core i7-9759H with a Nvidia GeForce GTX 1650 Max-Q.

To accurately predict the presence of anomalies based on the printing parameters during ongoing print jobs, the developed model and the developed Klipper module, Edna, must be integrated and work in synchrony.

To activate the prediction model of the edna module, a new parameter is introduced, signaling the module to disable the dataset creation functionality and instead predict using the captured images with the same image capturing mechanism. To enable the prediction mode the ***predict*** field on the edna module must be set to **True**, as shown in 3.22.

```

[edna]
ustreamer_url: http://localhost:8080/?action=stream
predict:True
snapshot_interval: 5
```

Figure 3.22: Edna module configuration for prediction mode.

The integration of the model network with the edna module requires the model network, including training checkpoints to be on the same directory as the edna module in order to be loaded and called to predict the input image parameters, this structure is depiceted in Figure 3.23

For the actual prediction capabilities, the existing capturing mechanism is used to input images onto the trained model, which then predicts the chosen printing parameters that relate to the presence of anomalies. To achieve prediction during ongoing prints, the image capturing mechanisms designed for the dataset creation is used to feed the model with new images, the subsequent prediction report for the each parameter is then exposed to the console, Figure 3.24.

Deep learning models, such as the models used in this research, require extensive computational capabilities due to their processing needs. The ODROID C4 SBC can run deep learning models with lower complexity, but struggles with higher complexity models. To address this concern, the models network is also running locally to predict the printing parameters, capturing images of the nozzle camera stream, exposed by the UStreamer video stream server.

Deep learning models, like those used in this research, require significant computational power due to their precessing needs. While the ODROID C4 SBC can handle deep learning model with low complexity, it struggles with higher complexity models has it doesn't have the processing requirements. To address this issue, the model's network is

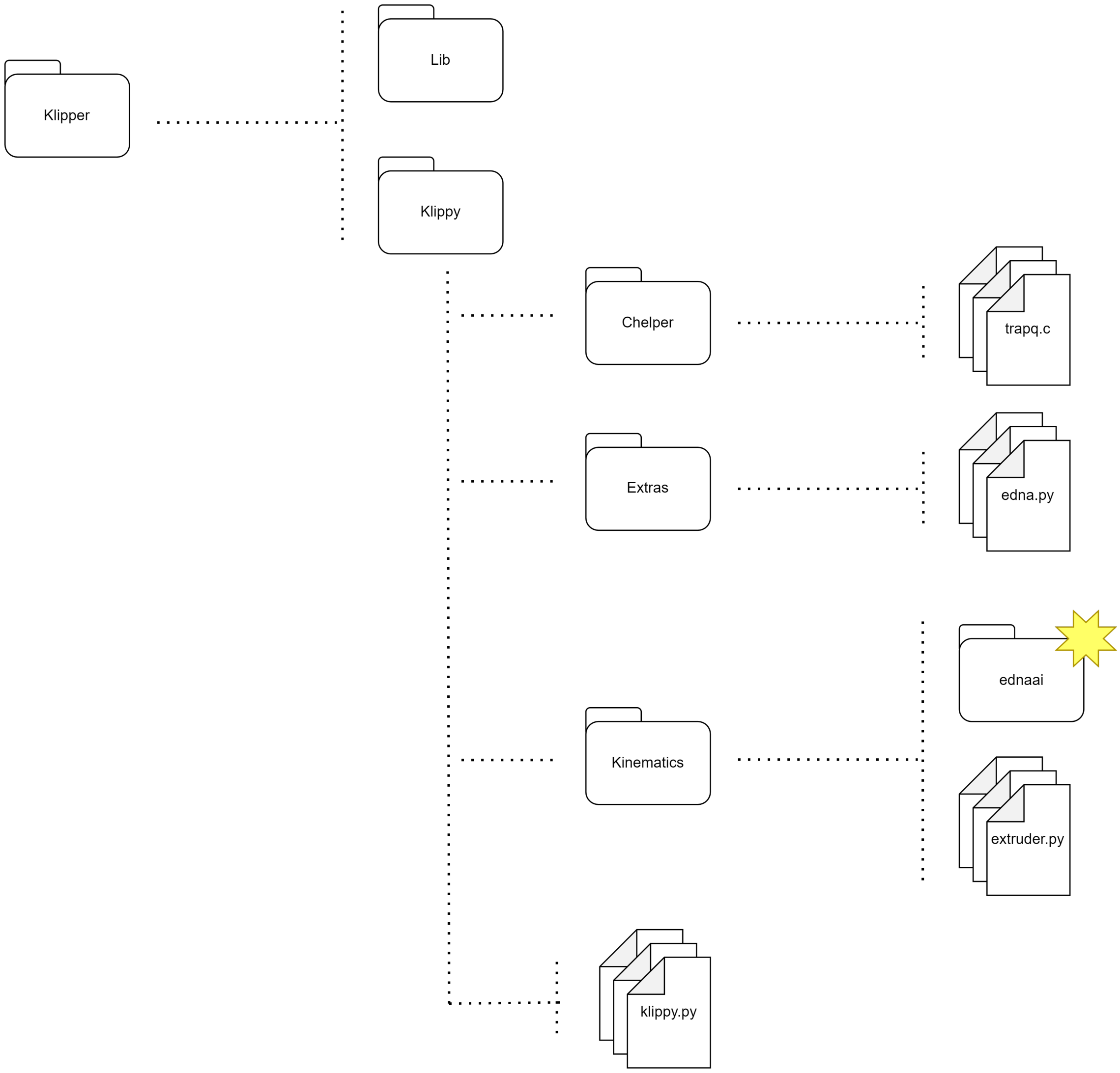


Figure 3.23: Klipper firmware software structure with the positioning of the models network to be used with edna module.

also running locally to predict the printing parameters, capturing images from the nozzle camera stream provided by the UStreamer video stream server. Locally, the model leverages a Nvidia GeForce GTX 1650 GPU to make predictions, although the computer has limited computational resources for running highly complex models, it should make inferences quicker.

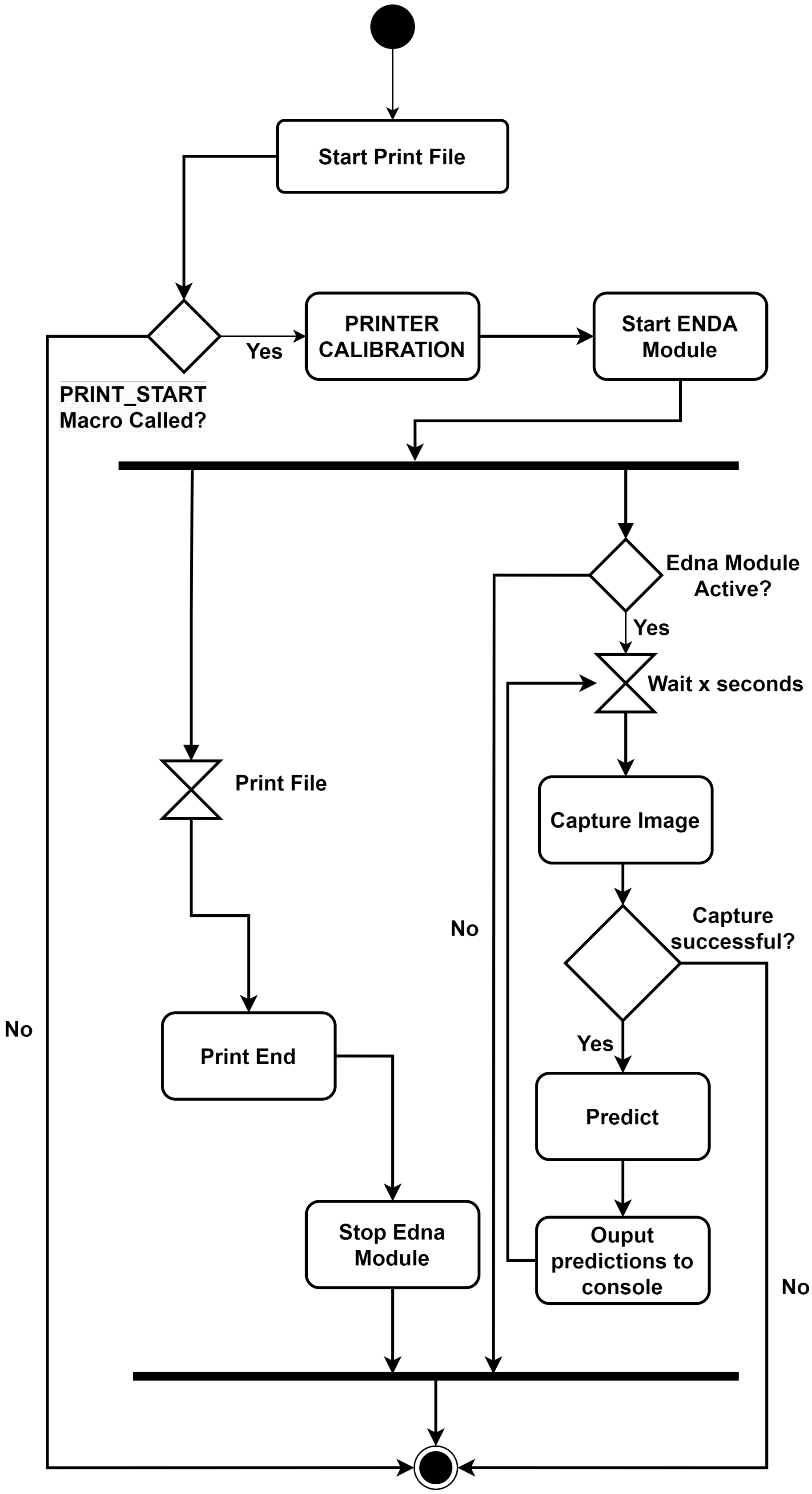


Figure 3.24: Activity diagram for the monitoring prototype.

4 Results

This chapter presents the results achieved from development of the proposed work, offering a detailed overview of the two highlighted contributions. The information is organized in the following sections:

- **Computer Vision Model Results** - This section details the results and technicalities from implementing the models introduced in subsection 3.5.3. It also reviews the challenges faced while developing the networks.

- **Dataset Results** - Details the outcomes from the produced dataset, providing an in-depth explanation of the problems encountered during the creation of the proposed dataset, explained in subsection 3.6.

4.1 Computer Vision Model Results

This section exposes the results from the modifications made to the Collaborative Autonomous Extrusion Network (CAXTON) network backbone, as detailed in subsection 3.5.3. Information is revealed in a structured manner, starting with the model's preparation for the training, validation, and testing phases. It then moves on to the results achieved and concludes with the challenges encountered during this development, as well as the final considerations.

4.1.1 Preparation

In order to test the effectiveness of the proposed modifications, detailed in subsection 3.5.3, with the original CAXTON model. The data utilized during the development phases of the models was the available dataset developed from Brion and Pattinson [7] (2022) as part of the study that produced the CAXTON model. The original CAXTON model and its respective dataset are considered to be the benchmark to which the development of new models should compare to.

Deep networks necessitate substantial processing power for training. One of the most significant challenges encountered during the model's training was not just the time-intensive nature of the process but also the limitations posed by the available hardware. Due to the lack of available processing power, the CAXTON dataset was reduced in this study in two stages:

- **Stage 1** - 6000 images where utilized to build the train, validation and predict datasets.
- **Stage 2** - 100,000 images where used to build the train, validation and predict datasets.

The train, validation and test datasets where split by 85%, 5% and 5% of the total amount of images available in each stage, respectively. For prediction an additional 29999 images were used from the entire dataset.

Data-augmentation techniques were utilized, this is done in order to increase the amount of varied data available on the training dataset, improve the models generalization abilities and robustness. This is especially important due to the reduction of the dataset, imposed by the limited processing power available. In this case, data is manipulated thorough common transformations including:

- Randomly rotating the image.
- Randomly scaling the image.
- Flipping the image, horizontally or vertically.
- Color jitter, includes changing the contrast, brightness, saturation and hue of the image.
- Normalization.

A crucial transformation done to input data is scaling the image to 224x224 dimensions, as this is the input size of the network.

All models require setting **hyper-parameters** before the training process begins, these parameters directly influence the performance and efficiency of the trained model. In this study, model training was achieved by using the following **hyper-parameters**:

1. **Precision**: Refers the numerical precision used during training and inference. It was set to 32.
2. **Batch Size**: Number of examples used on one iteration of the model (Epoch). Set to 32.
3. **Initial Learning rate**: Controls how much the models weights are changes. Set to $1e^{-4}$.

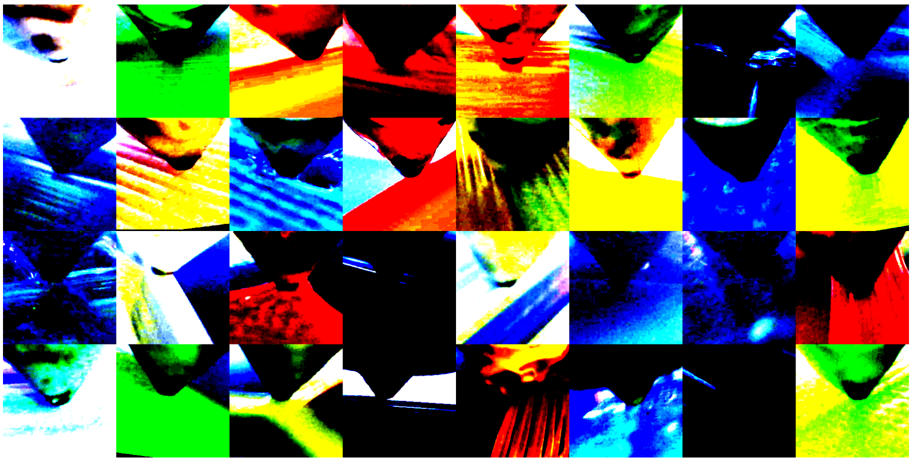


Figure 4.1: Example of transformations done to the dataset.

4. **Max Epochs**: The first training phase this parameter was set to 30. The last training phase where two models are validated, it was set to 50.

For **learning rate**, a scheduler was utilized, this adjust the learning rate during training, which can help the model to converge more quickly, and stabilize the training phase and increase the generalization of the model. The learning rate starts with a higher value, allowing the model to make larger updates on the weights. As the training phase progresses the value is reduced to fine-tune the model, this avoids overshooting. To this end, the class **ReduceLROnPlateu** was used as the networks scheduler, this scheduler decreases the learning rate by a factor of 0.1 when the metric stops improving. Another used optimization technique was the **AdamW** optimizer, this is a regularization technique to prevent the model from overfitting.

4.1.2 Training Stages

The initial training phase, utilizes 6,000 images from the whole CAXTON dataset, for the train, validation and test datasets.

All modified models exhibit slightly faster learning than the CAXTON model, this indicates that the modified models benefit from the introduction of a squeeze-and-excitation layer. The close alignment of the validation and training curves on the Accuracy graph, 4.2 suggest that the models do not suffer from overfitting issues and are capable of generalizing.

The modified se_after_residual model, exhibits a slight edge in comparison to the others, showing consistent performance both in the accuracy graph and loss graph.

Due to limited processing capabilities, the second phase of training was restricted to the CAXTON model and its variant, se_after_attention, which integrates Squeeze and Excitation Layer (SELayer)s after each residual unit after to each attention module. The purpose of this phase is to evaluate the performance of the original CAXTON model against its modified counterpart using an expanded dataset, here comprising 100,000 images.

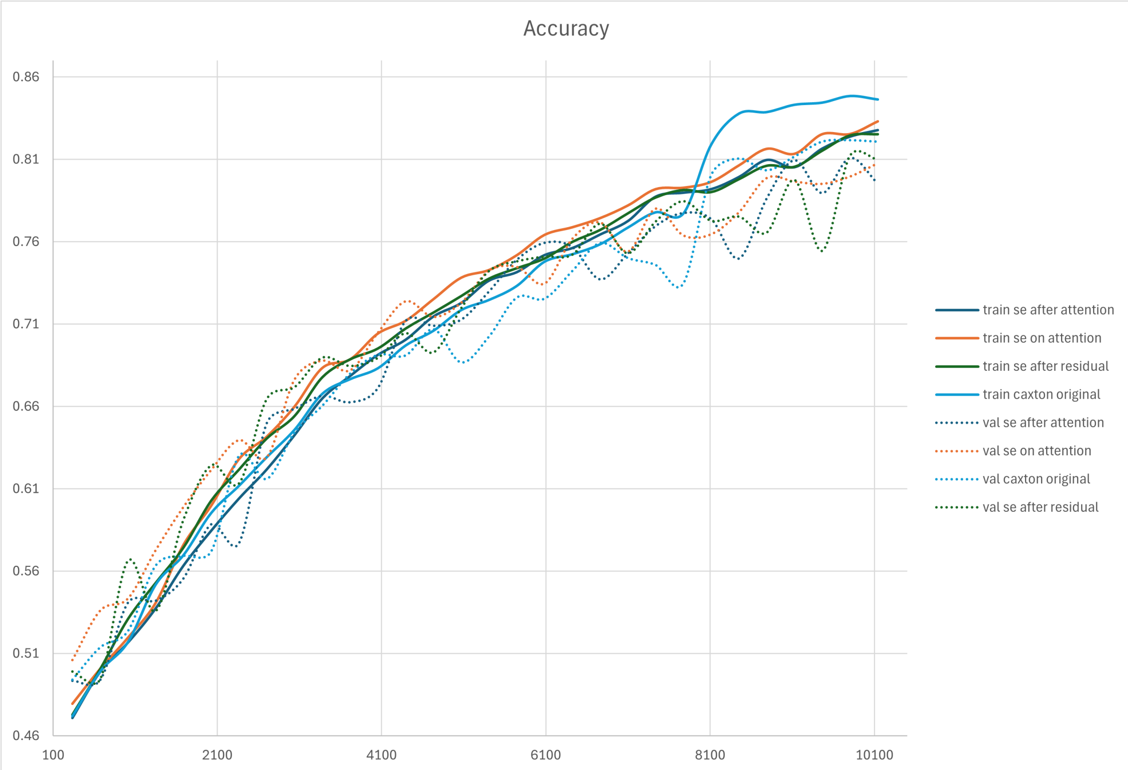


Figure 4.2: Training and validation Accuracy graph for all tested models.

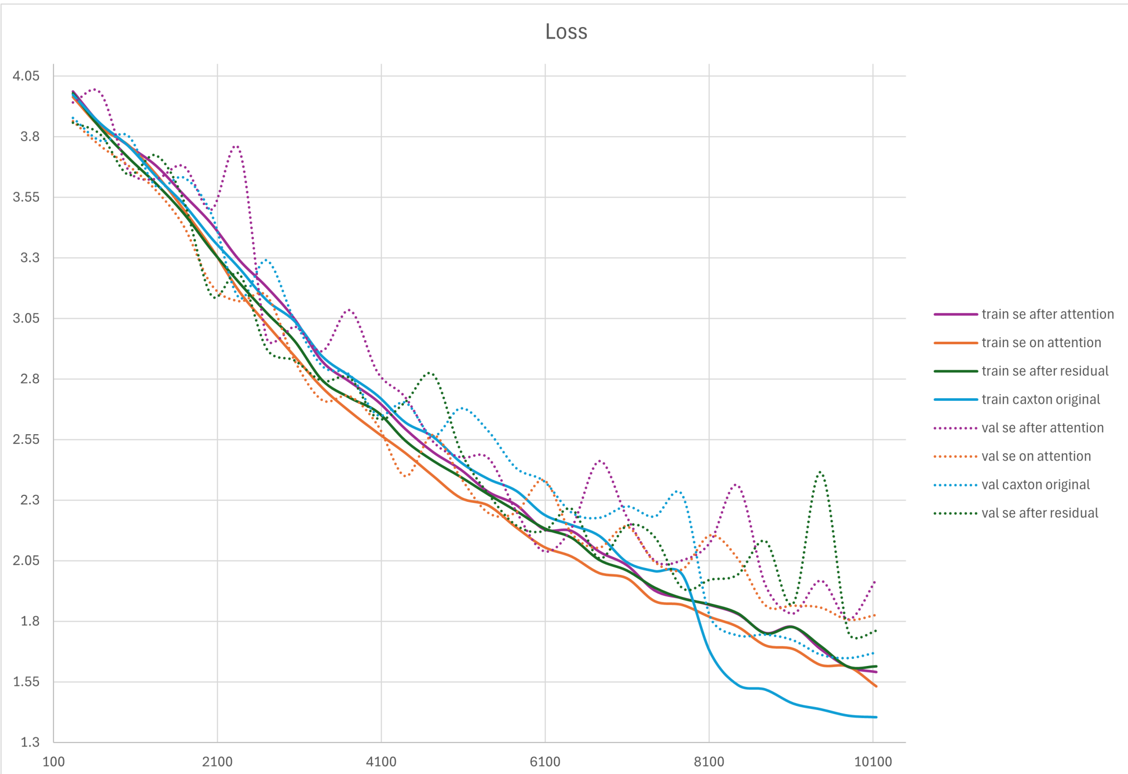


Figure 4.3: Training and validation Loss graph for all testes models.

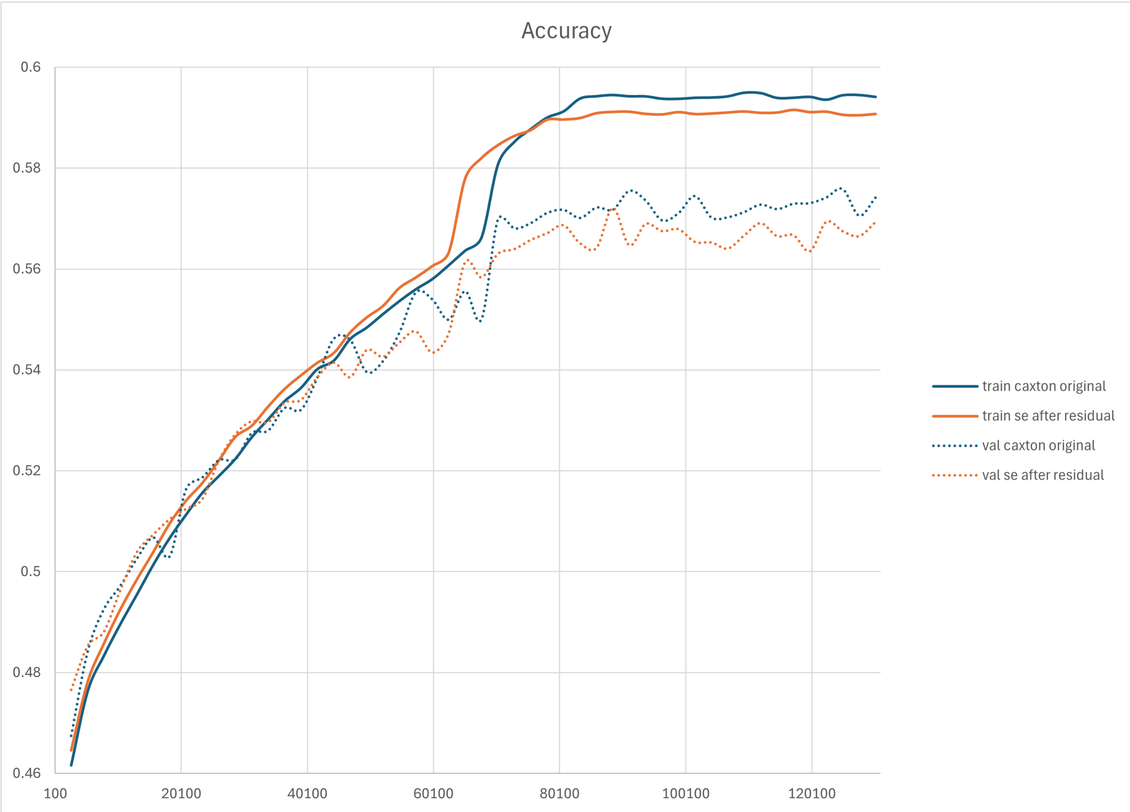


Figure 4.4: Caxton and modified withe SELayer after residual blocks accuracy graph.

The loss graph, Figure 4.5, shows that the train loss and validation loss both gradually decrease over training iterations. The validation loss on of the original model exhibits more fluctuations. On the other end, the modified model shows that its validation loss in generally lower than the original, this could indicate a better generalization performance. On the modified model, the training loss rapidly decreases in training, which suggests that the model has improved its learning efficiency.

The accuracy graph, Figure 4.4, displays that the initial phase of training, the CAXTON model shows a slightly faster increase in accuracy compared to the modified model (se_after_residual). This difference suggests that the original model learns more quickly on early stages of training compared to the modified model. On the other end, after this initial stage, the modified model se_after_attention, surpasses the original model, this indicates that the addition of a SELayer in the configuration detailed on 3.10, is effective at increasing the models abilities to learn faster and with higher accuracy.

At the end of training, the modified model plateaus sooner than the original model, this indicates that the model reached its peak performance sooner. Both models exhibit a similar accuracy, although the original model does surpass the modified one.

Two interpretations can be gathered from the accuracy graph:

1. The modified model exhibits a faster convergence than the original CAXTON model,

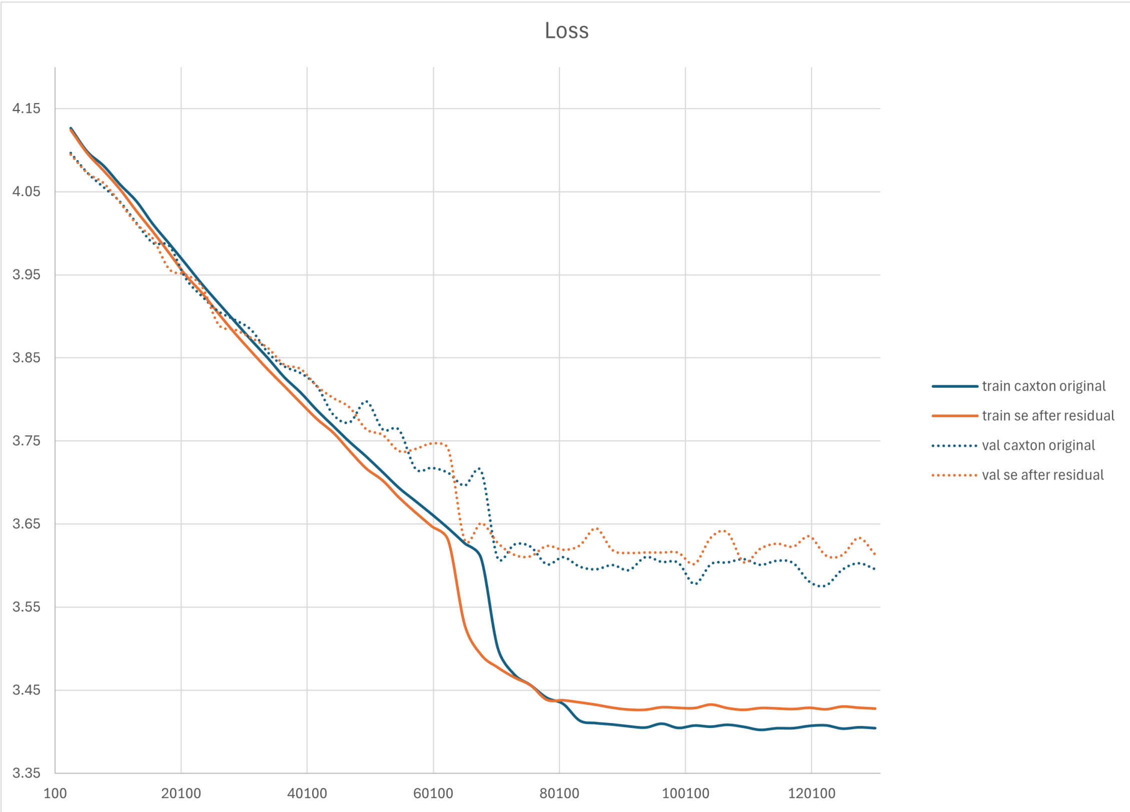


Figure 4.5: Caxton and modified with SELayer after residual blocks accuracy graph.

this likely indicates that the model configuration increases the models ability to focus on the most relevant features, leading it to learn more quickly.

2. The modified model se_after_attention, along the training procedure displays a more stable increase in accuracy, which can suggest that the modified model is more resistant from overfitting by highlighting important features and suppressing less relevant ones.

On the x-axis, representing the number of iterations, around the 65,000 mark, both models exhibit a peak in accuracy, this can be attributed to a decrease of the learning rate hyper-parameter. At this mark it should also be noted that the validation curve for both models does not follow the training curve, which could indicate issues with overfitting. This might be due to the subset used, consisting of 100,000 images from the entire CAXTON dataset. The subset could lack diversity and may not encompass the full spectrum of possibilities.

The complexity introduced by the introducing SELayers to the model does not significantly impact the modified model's training time, Table 4.1, only exhibiting a slight increase.

Table 4.1: Stage 2 training times

Model	Training time (days)
CAXTON	1.491
se_after_residual	1.738

Prediction testing was conducted utilizing an additional 29999 images from the entire CAXTON dataset. As a result of training both models with only a subset of the CAXTON dataset, the models perform poorly in real-world prediction tasks, with comparable results, the confusion matrices for each model are as follows:

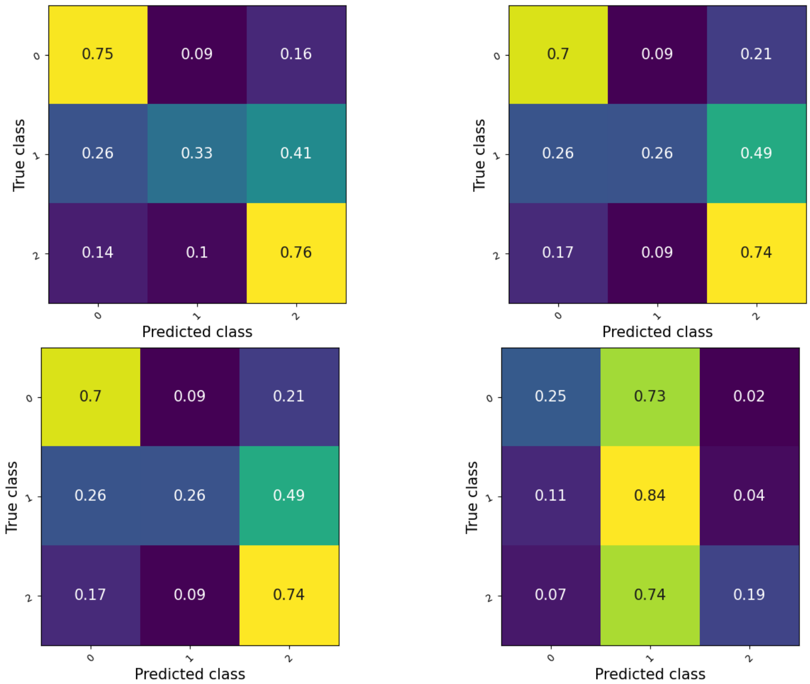


Figure 4.6: Confusion matrices for the CAXTON model, displayed for each head output as follows: ”flow rate” (top-left), ”feed rate” (top-right), ”z offset” (bottom-left) and ”hotend temperature” (bottom-right).

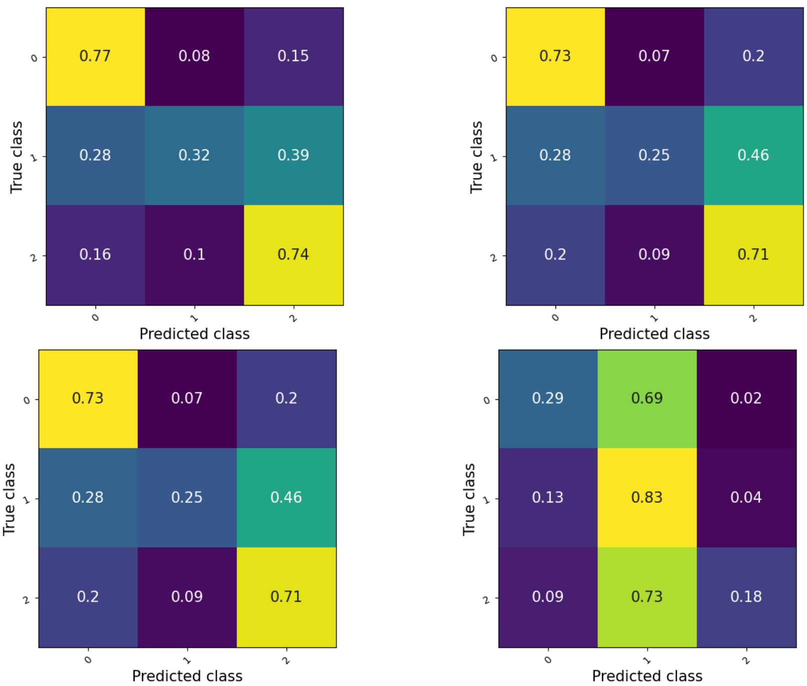


Figure 4.7: Modified model, se_after_residual, confusion matrices for each head output as follows: "flow rate" (top-left), "feed rate" (top-right), "z offset" (bottom-left) and "hotend temperature" (bottom-right).

4.1.3 Summary

One significant challenge, highlighted throughout the previous subsections, was the requirement for powerful processing capabilities. The lack of this requirement when training the models is considered to have impacted the final results, as the training was executed with relatively small sub-dataset.

After stage 2 of training, where both the original model and its modified counter part, with SELayers after each residual unit between the attention modules, the results indicate that the models posses similar performances. Although the modified model indicates that the changes made have effectively increased its performance, with improved stability and faster convergence. The overall performance of both models still remains comparable.

4.2 Dataset Results

This section initially exposes the results of the developed module for Klipper firmware, which enables the acquisition of filament deposition images and the creation of a dataset.

4.2.1 Data acquisition

The production of the dataset posed as a challenge, as many uncounted variables hindered the production of a more robust and diversified dataset. Firstly, the nozzle camera used is consists of a flat ribbon cable that connects the camera to a PCB, which automatically encodes the image. This ribbon cable is extremely delicate and easily damaged. Three nozzle cameras were necessary for the continuation of this work, which meant that the Three dimensional (3D) printer was not operational whenever a camera was broken.

For the production of the dataset, the printer is assumed to be extremely well calibrated, allowing no margin for error. The source of print anomalies should only be induced by the developed Klipper firmware module. Due to mechanically incorrect calibrations, such as the printer's bed plate losing its leveling, some prints were invalidated because the z offset value, which was assumed to be ″good″ initially, was actually either ″low″ or ″high″. This issue hindered some prints, as the automatic labeling was incorrect. The third nozzle camera used for this dataset had inferior quality when capturing images with movement, due to this reason some images are unusable because of the amount of noise.

The module developed for Klipper firmware, Edna, was successful in capturing the filament deposition images and aggregating all information necessary for the production of the dataset. For the production of the dataset anomalies were induced using Geometric Code (G-code) commands, this process was ultimately controlled solely by the developed module. Even thought hardships were encountered, 537,152 images of filament deposition were captured during the creation of the dateset, in 100 different prints.

The random nature of selecting which parameters to alter, also produces some unwanted results when creating a dataset that contains different combinations of parameters. In this dataset the parameters ″feed rate″, ″flow rate″, ″z offset″ and ″hotend temperature″ and ″bed temperature″ can have different combinations of ″low″, ″good″, ″high″. Every 10% of the total amount of the objects layers the Edna module randomizes which parameters to change via a G-code command. Despite the dataset containing over half a million captured images, many were ultimately unusable. When the z offset is changed to ″low″, the G-code command induces a decrease on the distance between the nozzle and the bed plate. This can be catastrophic for the printer, if the distance variation is to high, the nozzle can hit the bed plate and mechanically damage both the bed plate and the toolhead. In this case the printing job has to be stopped. Conversely, changing the z offset parameter from ″good″ to ″high″ involves sending a G-code command to increase the distance between the nozzle and the bed plate. This initial increase does not necessarily render the subsequent captured images invalid, but if the same type of parameter change is implemented again, the z offset becomes extremly high. After a certain point, the captured images become unusable, as the images only display stringing (filament is extruded into the air). This issue is evident in print number 27, which contained 32,129 filament deposition images. Most of the images captured in this print are invalid because the z offset parameter was set to "high" twice in a row. After filtering, 20,000 images were deleted as they provided no additional information. This left the print with 12,129 images, of which 9,007 contained information with different parameter combinations and 3,122 contained only stringing. These 3,122 images displaying stringing were retained in the dataset to provide the model with information about when the z offset parameter is "high". This type of filtering was applied to the entire dataset, resulting in 330,065 usable images.

An example of an image captured in this dataset is depicted on Figure 4.8. This figure exhibits a combination of different parameters, randomized by the Edna module.

Table 4.2: Labels for the image in Figure 4.8

Parameter	Parameter Value	Base value	Label (Numeric)	Label (Quantitative)
Feed rate	199%	100%	2	high
Flow rate	100%	100%	1	good
Z offset	0.038 mm	0 mm	2	high
Hotend temperature	262 ºC	240 ºC	2	high
Bed temperature	80 ºC	70 ºC	2	high

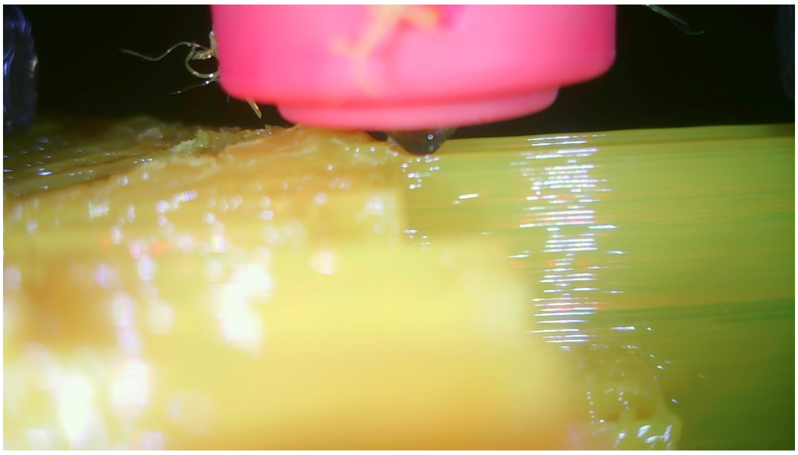


Figure 4.8: Example of a captured image from the dataset. Print number 3, image number 24425.

4.2.2 Dataset train results

To compare the dateset developed in this dissertation with the dateset from the CAXTON model, the models used in the last stage of training (see subsection 4.1.2) were retrained using the entire developed dataset.

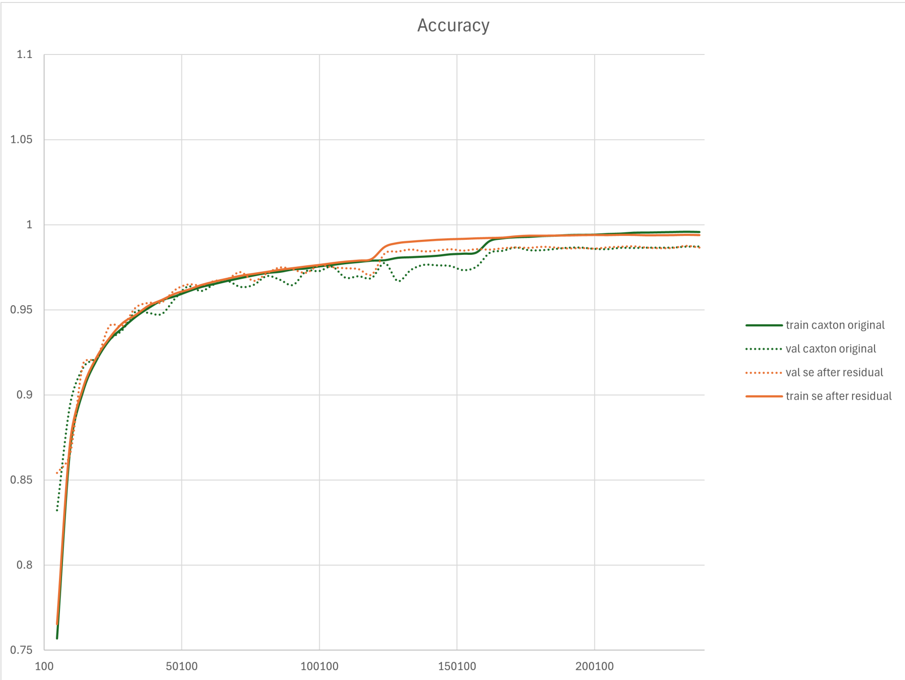


Figure 4.9: CAXTON and modified model with SELayer after residual blocks accuracy graph.

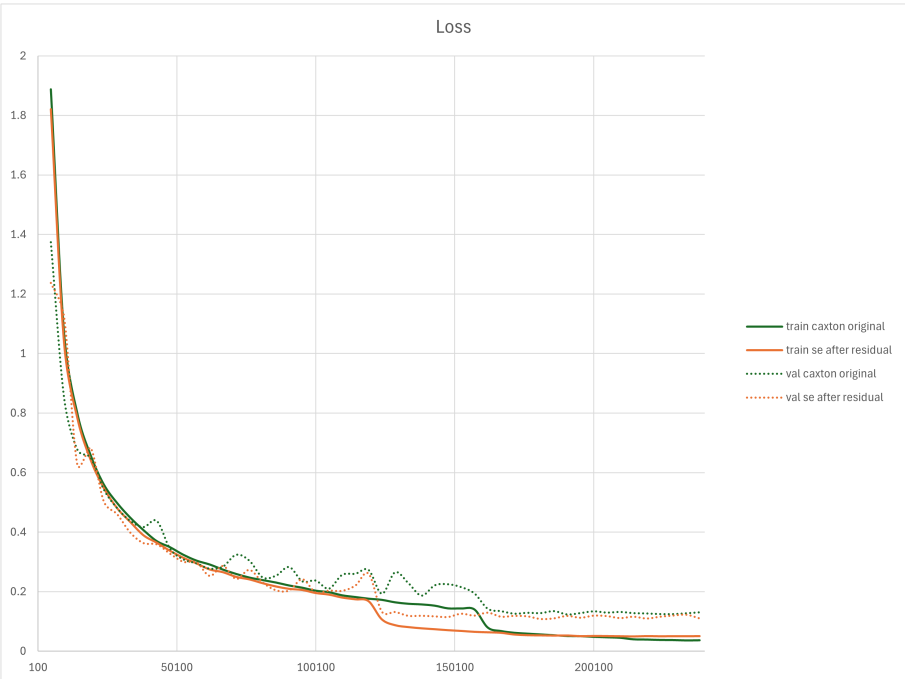


Figure 4.10: CAXTON and modified model with SELayer after residual blocks loss graph.

While the accuracy and loss graphs may suggest that the model was trained perfectly, the observations from the previous subsection imply that these graphs are not reliable indicators of the dataset's performance.

4.2.3 Summary

The development of the dataset posed as a challenge, not only due to mechanical failures which significantly delayed its production but also because of the randomness in selecting which parameters to alter. This randomness aspect resulted in a substantial number of unusable images. Another problem encountered is that the dataset does not adequately generalize across all parameter combinations, resulting in a dataset that lacks the necessary examples to effectively train the models proposed in this dissertation. Even with the encountered difficulties, the dataset contains 330,065 usable images (some of which contain noise), labeled accordingly.

4.3 Prototype Results

The integration between the deep learning network and the Klipper firmware Edna module was fairly straightforward, most of the image capturing mechanism was already implemented as part of the dataset creation portion of the work.

The ODROID C4 was able to capture filament deposition images and input them into the se_after_attention model version. The Single Board Computer (SBC) was able to predict two picture frames. The output for a single frame is depicted on Figure 4.11. Unfortunately, due to the already suspected lack of computational power from the SBC, Klipper firmware would consistently shutdown after two to three predictions, raising errors due to low resource availability, Figure 4.12. The local model prediction would consume to much processing power for Klipper to effectively control the printer and create predictions for the images.

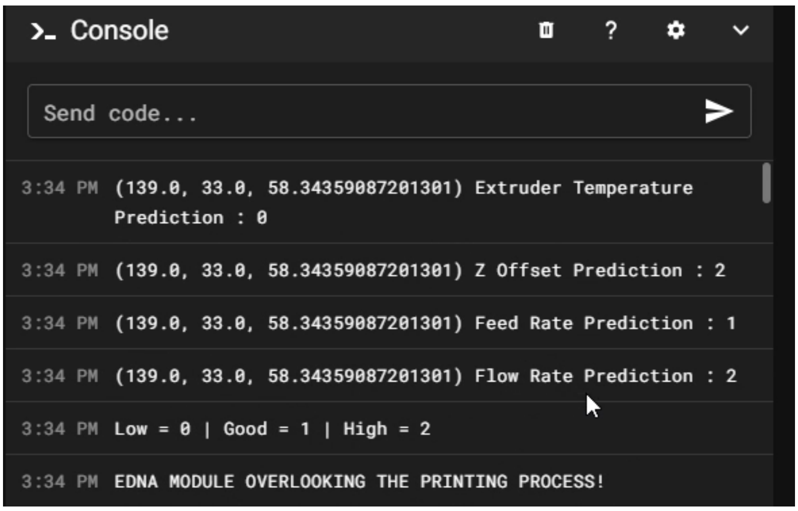


Figure 4.11: Local prediction using the Klipper firmware edna module.

Figure 4.12: Klipper shutdown error due to low resource availability due to the model requiring hight computational power.

Given the low computational power of the SBC for handling complex deep learning models, remote model prediction using the Nvidia GTX 1650 GPU produced better results. The output of the model prediction can be seen on Figure 4.13. This hardware was better suited for the task, allowing a different system to focus on the model without the additional burden of controlling the printer with Klipper firmware.

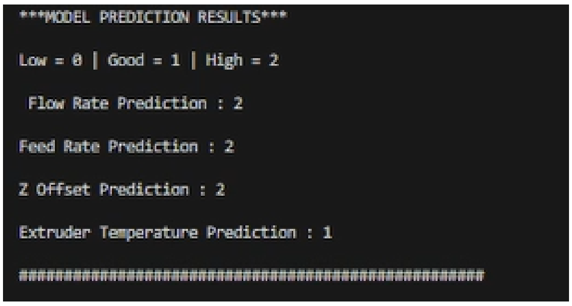


Figure 4.13: Example of the prediction output using remote monitoring.

4.3.1 Summary

Despite the model training not yielding the expected results due to limited computational resources, the monitoring prototypes remain useful, even if the inferred predictions by the trained model are inaccurate.

For the local prediction, the SBC requires more computational power to handle both the printer control and the deep learning model. The remote prediction separates these tasks between different systems, allowing the printer to focus on controlling the printer and another computer tasked with capturing and running the deep learning model predictions.

5

Conclusion & Future Work

Additive manufacturing methods are increasingly gaining widespread popularity from both manufacturing industries as from the consumer market in the Do it yourself (DIY) space. In the demanding era of adaptability and rapid development, fused deposition modeling Three dimensional (3D) printers pose as a respectable manufacturing method that offers both flexibility and cost-effective solutions for the set requirements. Current quality assessment methods continue to heavily rely on extensive human inspection which can be both flawed and destructive. Leveraging Deep Learning (DL) architectures to evaluate images for the presence of anomalies demonstrates exceptional results.

This dissertation was divided into two main contributions, each fundamental to achieve the outlines objective, Section 1.2. The first contribution being the creation of a imaging dataset which meant to be a complement to the existing CAXTON dataset while, at the same time, create room way for possible future requirements. In this development stage the it was created a module that can be integrated within Klipper firmware, enabling capturing images of filament deposition during an ongoing print. The module developed only requires a printer equipped with a camera and a nozzle camera, rendering it accessible to anyone who intends to contribute with images of their prints. The created dataset offered backward compatibility with prior work and only added new labels that are theorized to be necessary in the future. More than 500,000 images where collected during this stage by utilizing a modified Creality Ender 3 printer. Despite this contribution, numerous challenges were encountered. Due to the random nature of generating automatic labels for printer parameters, the first problem relates to the amount of images that were rendered unusable. This aspect hindered the quantity of data available within the dataset, causing the final dataset to only be comprised of about 300,000 images. The second problem stems from the lack of diversity on the dataset. The dataset contains more combinations of parameters of one type than others, this not only decreased the amount of examples for the models to learn from, but also causes the data to be imbalanced.

Due to hardware issues, the camera was replaced three times. The final camera used was of lower quality than the previous two, and it was this camera that was used for the majority of the dataset. The lower quality introduced noise into the dataset, which is not inherently a disadvantage, however in this dataset, it further reduced the dataset quality. Additionally, due to time constraints, further testing necessary to validate the quality of the produced dataset was not conducted.

Training both the Collaborative Autonomous Extrusion Network (CAXTON) model and the modified version with Squeeze and Excitation Layer (SELayer)s after each residual unit following the attention modules, resulted on the models to plateau quickly, demonstrating that the model does learn from the data. But given the previous considerations it is fair to assume that the model is learning misleading information and the resulting graphs should not be considered as indicators that the dataset is good.

Regarding the second major contribution, it should be noted that the training phase of the proposed model enhancements faced significant challenges due to the limited computational power available and the use of a highly complex Multi-head neural network. Three different methods were employed to train the proposed modified networks: Google Colab, which frequently resulted in kernel timeouts during training; Google Cloud Vertex AI Notebooks, which provided higher processing power but was not feasible for extensive testing due to the cost of allocated resources; and local training, which offered the lowest processing power, but provided the flexibility necessary for testing all models. Local processing was selected as the method to train all models in this dissertation for the reasons previously mentioned. Consequently, model training was conducted using only a small subset of images from the entire CAXTON dataset, as training with the entire dataset resulted in training times on the order of months. The subsets utilized for training were divided into two stages: Stage 1 consisted of training all models with 6,000 images from the entire CAXTON dataset; Stage 2 involved selecting one modified model, and both the original CAXTON model and the selected modified model were trained with 100,000 images. The CAXTON model was trained in both scenarios as it serves as the benchmark. The suggested modification involved integrating a Squeeze-and-Excitation Layer (SELayer) into the CAXTON backbone to enhance the model's attention capabilities. This integration allows the model to highlight important features of the input image while simultaneously suppressing less relevant ones, thereby increasing the model's performance. Along with this integration, three distinct configurations were proposed. The first configurations placed the SELayer after each residual unit following each attention modules. On the other two configurations, the SELayer was inserted into the basic building block of the networks, the residual unit. The second proposed model model configuration proposed replacing the residual units following the attention models with its modified counterpart. The third and last proposal, modified every residual unit inside each attention module with the modified residual unit.

The results from the first training stage suggest that all proposed modifications are effective at enhancing the model's ability to focus on important features of the input image, while simultaneously increasing the training stability, performance and generalization abilities. Although at the end training stage the original CAXTON model does exhibit a slight advantage, the modified models are comparable. The trained models do not plateau, suggesting that there is potential to train each model further. On the second training phase, both the CAXTON model and the model with the SELayer after each residual unit following the attention modules were trained with an increased amount of data and with increased iterations. The results indicate once again that the modified model benefits from the addition of a squeeze-and-excitation layer, demonstrating quicker learning, better generalization and efficiency. The findings suggest that the learning process benefits from the integration of SELayers. Nevertheless, in terms of accuracy, both the modified models and the CAXTON remain comparable, yielding similar results.

Although, the produced dataset does contain good images, which can be used to increment the CAXTON dataset, there is a crucial problem that needs to be addressed in future work. The dataset lacks a sufficient variety of examples to cover the combination of problems adequately. To address this issue, it is proposed to produce more images from prints with different combinations of problems. Another proposed enhancement is to utilize different commercial printers to increase the diversity of the dataset images.

The findings from training the modified models indicate that the introduction of an SELayer onto the CAXTON backbone does increase its learning capabilities. One major complication with testing these modified models, was the lack of available processing power. This deficit resulted in training all models with only a subset of the entire CAXTON dataset. Future work should mainly focus on training the modified models with a bigger amounts of data, in this case with the entirety of the CAXTON dataset. Testing the models with a larger dataset is considered essential for further validating the performance of the modified models. Additionally, since SELayer is an attention mechanism, future work should also focus on testing additional attention mechanisms on the CAXTON model backbone in order to increase the model's performance and learning capabilities. This recommendation was initiated and the code developed is available on Github: https://github.com/HugoCLSC/EDNA .

The recommendations previously mentioned are believed to be essential in the creation of future state-of-the-art models that do not lack data resources and exhibit good performance when applied to anomaly detection in Fused Deposition Modeling (FDM) 3D printers.

The current work, was developed with the assistance of **Blockstec**, contributing not only with valuable expertise on FDM 3D printers but also with hardware required for the construction of the dataset.

Bibliography

[1] *3D Printer G-code Commands: Main List & Quick Tutorial | All3DP*. Accessed 17-09-2024. URL: https://all3dp.com/2/3d-printer-g-code-commands-list-tutorial/ (cit. on p. 10).

[2] Osama Abdulhameed et al. "Additive manufacturing: Challenges, trends, and applications". In: *Advances in Mechanical Engineering* 11.2 (2019-02). ISSN: 16878140. DOI: 10.1177/1687814018822880 (cit. on pp. 9–13).

[3] Masoumeh Aminzadeh and Thomas R. Kurfess. "Online quality inspection using Bayesian classification in powder-bed additive manufacturing from high-resolution visual camera images". In: *Journal of Intelligent Manufacturing* 30.6 (2019-08), pp. 2505–2523. ISSN: 15728145. DOI: 10.1007/s10845-018-1412-0 (cit. on pp. 28, 35).

[4] Filipe Bruno André Ferreira. "Computer vision-based quality inspection for additive manufacturing". MA thesis. NOVA School of Science Technology, NOVA University Lisbon, 2023-12. URL: http://hdl.handle.net/10362/167929 (cit. on pp. 25, 26).

[5] *Artificial Intelligence (AI) vs. Machine Learning | Columbia AI*. Accessed 21-09-2024. URL: https://ai.engineering.columbia.edu/ai-vs-machine-learning/ (cit. on p. 19).

[6] Anjali Bhardwaj. *What is a Perceptron? – Basics of Neural Networks | by Anjali Bhardwaj | Towards Data Science*. Accessed 22-09-2024. URL: https://towardsdatascience.com/what-is-a-perceptron-basics-of-neural-networks-c4cfea20c590 (cit. on p. 22).

[7] Douglas A.J. Brion and Sebastian W. Pattinson. "Generalisable 3D printing error detection and correction via multi-head neural networks". In: *Nature Communications* 13.1 (2022-12). ISSN: 20411723. DOI: 10.1038/s41467-022-31985-y (cit. on pp. 32, 36, 38, 44, 46, 53, 54, 67).

[8] Bitkom eV Bundesverband Informationswirtschaft et al. *Implementation Strategy Industrie 4.0*. Tech. rep. Bitkom e.V, VDMA e.V, ZVEI e.V, 2016. URL: https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2016/januar/Implementation_Strategy_Industrie_4.0_-_Report_on_the_results_of_Industrie_4.0_Platform/Implementation-Strategy-Industrie-40-ENG.pdf%20www.bitkom.org%20www.vdma.org%20www.zvei.org (cit. on pp. 6–8).

[9] *CIFAR-10 and CIFAR-100 datasets*. Accessed 21-09-2024. URL: https://www.cs.toronto.edu/~kriz/cifar.html (cit. on p. 19).

[10] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)". In: (2015-11). URL: http://arxiv.org/abs/1511.07289 (cit. on p. 23).

[11] Jeremy Cohen. *Computer Vision applications in Self-Driving Cars | by Jeremy Cohen | Becoming Human: Artificial Intelligence Magazine*. Accessed 22-09-2024. URL: https://becominghuman.ai/computer-vision-applications-in-self-driving-cars-610561e14118 (cit. on p. 25).

[12] *CoreXY - RepRap*. Accessed 16-08-2024. URL: https://reprap.org/wiki/CoreXY (cit. on p. 18).

[13] Ugandhar Delli and Shing Chang. "Automated Process Monitoring in 3D Printing Using Supervised Machine Learning". In: vol. 26. Elsevier B.V., 2018, pp. 865–870. DOI: 10.1016/j.promfg.2018.07.111. URL: https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichhart-reference_architectural_model_industrie_4.0_rami_4.0.pdf (cit. on pp. 27, 35).

[14] *Delta geometry - RepRap*. Accessed 16-08-2024. URL: https://reprap.org/wiki/Delta_geometry (cit. on p. 17).

[15] Duarte Dias Bragadesto. "Nutritional Value extraction of food exploiting computer vision and near infrared Spectrometry". MA thesis. NOVA School of Science Technology, NOVA University Lisbon, 2019-02. URL: http://hdl.handle.net/10362/115841 (cit. on pp. 20, 22, 25).

[16] André Dionísio Bettencourt da Silva Parreira Rocha. "Increase the adoption of Agent-based Cyber-Physical Production Systems through the Design of Minimally Invasive Solutions". PhD thesis. Faculdade Ciências e Tecnonogiada Universidade NOVA de Lisboa, 2018-07. DOI: http://hdl.handle.net/10362/58083. URL: http://hdl.handle.net/10362/58083 (cit. on pp. 6–8).

[17] Pedro Domingos and Michael Pazzani. "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss". In: *Machine Learning* 29.2-3 (1997). ISSN: 08856125. DOI: 10.1023/a:1007413511361 (cit. on p. 20).

[18] Michaela T. Espino et al. *Statistical methods for design and testing of 3D-printed polymers*. 2023. DOI: 10.1557/s43579-023-00332-7 (cit. on p. 35).

[19] Guilherme Estima de Oliveira Lemos. "Métodos de visão computadorizada para monitorização do crescimento de pera rocha". MA thesis. NOVA School of Science Technology, NOVA University Lisbon, 2020-11 (cit. on pp. 22, 24–26).

[20] *Everything You Need to Know About 3D Printer Heat Break - 3D Insider*. Accessed 16-09-2024. URL: https://3dinsider.com/3d-printer-heat-break/ (cit. on p. 15).

[21] Pranav Gharge. *Klipper Firmware: 6 Key Reasons to Use It for Your 3D Printer | Clever Creations*. Accessed 10-09-2024. URL: https://clevercreations.org/klipper-firmware-3d-printing-printer/ (cit. on p. 43).

[22] Christian Gobert et al. "Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging." In: *Additive Manufacturing* 21 (2018). ISSN: 22148604. DOI: 10.1016/j.addma.2018.04.005 (cit. on p. 28).

[23] Steve Graves. "Johann C. Rocholl (Rostock) Style Delta Robot Kinematics". In: (). Accessed 14-8-2024. URL: https://fab.cba.mit.edu/classes/863.15/section.CBA/people/Spielberg/Rostock_Delta_Kinematics_3.pdf (cit. on p. 17).

[24] Abid Haleem and Mohd Javaid. *Additive manufacturing applications in industry 4.0: A review*. 2019-12. DOI: 10.1142/S2424862219300011 (cit. on pp. 9, 11).

[25] Chris Harding, Franek Hasiuk, and Aaron Wood. "TouchTerrain-3D printable terrain models". In: *ISPRS International Journal of Geo-Information* 10.3 (2021-03). ISSN: 22209964. DOI: 10.3390/ijgi10030108 (cit. on p. 14).

[26] Sonia Holland, Chris Tuck, and Tim Foster. "Selective recrystallization of cellulose composite powders and microstructure creation through 3D binder jetting". In: *Carbohydrate Polymers* 200 (2018). ISSN: 01448617. DOI: https://doi.org/10.1016/j.carbpol.2018.07.064 (cit. on p. 13).

[27] Jie Hu et al. "Squeeze-and-Excitation Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.8 (2020). ISSN: 19393539. DOI: 10.1109/TPAMI.2019.2913372 (cit. on pp. 48, 50).

[28] *ImageNet*. Accessed 21-09-2024. URL: https://image-net.org/ (cit. on p. 19).

[29] Mohammadreza Iman, Hamid Reza Arabnia, and Khaled Rasheed. *A Review of Deep Transfer Learning and Recent Advancements*. 2023. DOI: 10.3390/technologies11020040 (cit. on p. 21).

[30] Jordan Inturrisi et al. *Piecewise Linear Units Improve Deep Neural Networks*. 2021. arXiv: 2108.00700 [cs.LG]. URL: https://arxiv.org/abs/2108.00700 (cit. on p. 23).

[31] Christian Janiesch, Patrick Zschech, and Kai Heinrich. "Machine learning and deep learning". In: *Electronic Markets* 31.3 (2021). ISSN: 14228890. DOI: 10.1007/s12525-021-00475-2 (cit. on pp. 19–21, 24–26).

[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". In: *Advanced Intelligent Systems* 2.1 (2020-01). ISSN: 2640-4567. DOI: 10.1002/aisy.201900130 (cit. on p. 31).

[33] Zeqing Jin et al. *Machine Learning for Advanced Additive Manufacturing*. 2020-11. DOI: 10.1016/j.matt.2020.08.023 (cit. on pp. 19, 35).

[34] Vaibhav Kadam et al. "Enhancing surface fault detection using machine learning for 3d printed products". In: *Applied System Innovation* 4.2 (2021-06). ISSN: 25715577. DOI: 10.3390/asi4020034 (cit. on pp. 20, 28).

[35] lle Kauppila. *SLS 3D Printer Buyer's Guide 2024 | All3DP Pro*. Accessed 20-09-2024. URL: https://all3dp.com/1/best-sls-3d-printer-desktop-industrial/ (cit. on p. 13).

[36] Hyungjung Kim, Hyunsu Lee, and Sung Hoon Ahn. "Systematic deep transfer learning method based on a small image dataset for spaghetti-shape defect monitoring of fused deposition modeling". In: *Journal of Manufacturing Systems* 65 (2022-10), pp. 439–451. ISSN: 02786125. DOI: 10.1016/j.jmsy.2022.10.009 (cit. on pp. 21, 30).

[37] Hironori Kondo. *What Is PLA? – 3D Printing Materials Simply Explained | All3DP*. Accessed 17-09-2024. 2019-05. URL: https://all3dp.com/2/what-is-pla-3d-printing-materials-simply-explained/ (cit. on p. 39).

[38] Alex Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. Tech. rep. 2009 (cit. on p. 19).

[39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Communications of the ACM* 60.6 (2017-06), pp. 84–90. ISSN: 15577317. DOI: 10.1145/3065386 (cit. on pp. 19, 29).

[40] Ohyung Kwon et al. "A deep neural network for classification of melt-pool images in metal additive manufacturing". In: *Journal of Intelligent Manufacturing* 31.2 (2020). ISSN: 15728145. DOI: 10.1007/s10845-018-1451-6 (cit. on p. 30).

[41] *Linear Delta Printer Kinematics - RepRap*. Accessed 16-09-2024. URL: https://reprap.org/wiki/Linear_Delta_Printer_Kinematics (cit. on p. 17).

[42] João M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/main/template.pdf (cit. on p. i).

[43] Yang Lu. "Industry 4.0: A survey on technologies, applications and open research issues". In: *Journal of Industrial Information Integration* 6 (2017-06), pp. 1–10. ISSN: 2452414X. DOI: 10.1016/j.jii.2017.04.005 (cit. on pp. 5, 6).

[44] Jiaqi Lyu and Souran Manoochehri. "Online Convolutional Neural Network-based anomaly detection and quality control for Fused Filament Fabrication process". In: *Virtual and Physical Prototyping* 16.2 (2021). ISSN: 17452767. DOI: 10.1080/17452759.2021.1905858 (cit. on p. 35).

[45] José Manuel Matos Ribeiro Da Fonseca. *Redes Neuronais*. Lecture slides. Unpublished material. 2023 (cit. on pp. 22–24, 26).

[46] Rajshree Mathur. "3D Printing in Architecture". In: *IJISET-International Journal of Innovative Science, Engineering & Technology* 3.7 (2016-07), pp. 583–591. ISSN: 2348-7968. URL: www.ijiset.com (cit. on p. 12).

[47] Bruna Driussi Mistro Matos et al. "Evaluation of commercially available polylactic acid (PLA) filaments for 3D printing applications". In: *Journal of Thermal Analysis and Calorimetry* 137.2 (2019). ISSN: 15882926. DOI: 10.1007/s10973-018-7967-3 (cit. on p. 39).

[48] Yutaka Matsuo et al. "Deep learning, reinforcement learning, and world models". In: *Neural Networks* 152 (2022). ISSN: 18792782. DOI: 10.1016/j.neunet.2022.03.037 (cit. on p. 21).

[49] Samreen Naeem et al. "An Unsupervised Machine Learning Algorithms: Comprehensive Review". In: *International Journal of Computing and Digital Systems* 13.1 (2023), pp. 911–921. ISSN: 2210142X. DOI: 10.12785/ijcds/130172 (cit. on pp. 19, 20).

[50] Gaurav Nair. *Activation Function in Neural Networks: Sigmoid, Tanh, ReLU, Leaky ReLU, Parametric ReLU, ELU, Softmax, GeLU | Medium*. Accessed 24-09-2024. URL: https://medium.com/@gauravnair/the-spark-your-neural-network-needs-understanding-the-significance-of-activation-functions-6b82d5f27fbf (cit. on p. 22).

[51] Nik. *Softmax Activation Function for Deep Learning: A Complete Guide • datagy*. Accessed 22-09-2024. URL: https://datagy.io/softmax-activation-function/ (cit. on p. 22).

[52] Jackson O'Connel. *The Types of FDM 3D Printers: Cartesian, CoreXY & More | All3DP*. Accessed 21-09-2024. URL: https://all3dp.com/2/cartesian-3d-printer-delta-scara-belt-corexy-polar/ (cit. on pp. 17, 18).

[53] Benedict O'Neill. *PETG Print Settings: Adjusting Temperature, Speed & Retraction to Improve Printing*. Accessed 28-09-2024. URL: https://www.wevolver.com/article/petg-print-settings (cit. on p. 39).

[54] Keiron O'Shea and Ryan Nash. "An Introduction to Convolutional Neural Networks". In: (2015-11). URL: http://arxiv.org/abs/1511.08458 (cit. on pp. 25, 26).

[55] Panzarkatten. *HEX Remix mod of 3DO Nozzle camera mount for Voron Stealthburner by Panzarkatten | Download free STL model | Printables.com*. Accessed 22-08-2024. 2023-03. URL: https://www.printables.com/model/381456-hex-remix-mod-of-3do-nozzle-camera-mount-for-voron (cit. on p. 42).

[56] Aliaksei L. Petsiuk and Joshua M. Pearce. "Open source computer vision-based layer-wise 3D printing analysis". In: *Additive Manufacturing* 36 (2020-12). ISSN: 22148604. DOI: 10.1016/j.addma.2020.101473 (cit. on pp. 36, 37).

[57] B. A. Praveena et al. "A comprehensive review of emerging additive manufacturing (3D printing technology): Methods, materials, applications, challenges, trends and future potential". In: *Materials Today: Proceedings*. Vol. 52. Elsevier Ltd, 2022, pp. 1309–1313. DOI: 10.1016/j.matpr.2021.11.059 (cit. on pp. 11, 13).

[58] *Print Quality Guide | Simplify3D Software*. Accessed 23-09-2024. URL: https://www.simplify3d.com/resources/print-quality-troubleshooting/ (cit. on p. 36).

[59] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. "Searching for Activation Functions". In: (2017-10). URL: http://arxiv.org/abs/1710.05941 (cit. on pp. 22, 23).

[60] Mark Rehorst. *Mark Rehorst's Tech Topics: CoreXY Mechanism Layout and Belt Tensioning*. Accessed 21-09-2024. URL: https://drmrehorst.blogspot.com/2018/08/corexy-mechanism-layout-and-belt.html (cit. on p. 18).

[61] Vasja Roblek, Maja Meško, and Alojz Krapež. "A Complex View of Industry 4.0". In: *SAGE Open* 6.2 (2016-06). ISSN: 21582440. DOI: 10.1177/2158244016653987 (cit. on pp. 5, 6).

[62] Roumiana Yossifova Ilieva and Mario Angelov Nikolov. "The Impact of Machine Vision in Additive Manufacturing". In: *Proc. XI National Conference with International Participation "Electronica 2020"* 1.1 (2020-09). ISSN: 25044494. DOI: 10.3390/jmmp1010002 (cit. on p. 25).

[63] Olga Russakovsky et al. "ImageNet Large Scale Visual Recognition Challenge". In: (2014-09). URL: http://arxiv.org/abs/1409.0575 (cit. on p. 19).

[64] Karsten Scheweichhart. *The Internet of Things and Services Graphics © Bosch Rexroth AG*. Tech. rep. Plattform Industrie 4.0 (cit. on p. 8).

[65] Carolyn Schwaar. *The Complete Guide to Nylon 3D Printing | All3DP Pro*. Accessed 28-09-2024. 2024-07. URL: https://all3dp.com/2/nylon-3d-printing-how-to-get-nylon-3d-printed/ (cit. on p. 39).

[66] Luke Scime and Jack Beuth. "A multi-scale convolutional neural network for autonomous anomaly detection and classification in a laser powder bed fusion additive manufacturing process". In: *Additive Manufacturing* 24 (2018-12), pp. 273–286. ISSN: 22148604. DOI: 10.1016/j.addma.2018.09.034 (cit. on pp. 29, 35).

[67] Luke Scime and Jack Beuth. "Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm". In: *Additive Manufacturing* 19 (2018). ISSN: 22148604. DOI: 10.1016/j.addma.2017.11.009 (cit. on p. 29).

[68] *Selective Laser Sintering - SLS 3D Print Service - 3D Print Technology*. Accessed 20-09-2024. URL: https://rapidfab.ricoh-europe.com/technologies/selective-laser-sintering/ (cit. on p. 12).

[69] Syed Imran Shafiq et al. "Virtual engineering object/virtual engineering process: A specialized form of cyber physical system for industrie 4.0". In: *Procedia Computer Science*. Vol. 60. 1. 2015. DOI: 10.1016/j.procs.2015.08.166 (cit. on pp. 6–8).

[70] Anthony Simons, Kossi L.M. Avegnon, and Cyrus Addy. "Design and development of a delta 3D printer using salvaged e-waste materials". In: *Journal of Engineering (United Kingdom)* 2019 (2019). ISSN: 23144912. DOI: 10.1155/2019/5175323 (cit. on p. 18).

[71] Matt Tyson. *Advanced Guide to printing ABS Filament | User Guides*. Accessed 28-09-2024. 2018-06. URL: https://www.3dprintingsolutions.com.au/User-Guides/how-to-3d-print-abs-filament (cit. on p. 39).

[72] Shahadat Uddin et al. "Comparing different supervised machine learning algorithms for disease prediction". In: *BMC Medical Informatics and Decision Making* 19.1 (2019). ISSN: 14726947. DOI: 10.1186/s12911-019-1004-8 (cit. on pp. 20, 21).

[73] Birgit Vogel-Heuser and Dieter Hess. *Guest Editorial Industry 4.0-Prerequisites and Visions*. 2016-04. DOI: 10.1109/TASE.2016.2523639 (cit. on pp. 6, 8).

[74] *VORON Design*. Accessed 25-09-2024. URL: https://www.vorondesign.com/ (cit. on pp. 10, 42).

[75] Fei Wang et al. "Residual attention network for image classification". In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*. Vol. 2017-January. 2017. DOI: 10.1109/CVPR.2017.683 (cit. on pp. 44, 47).

[76] *Welcome - Klipper documentation*. Accessed 25-09-2024. URL: https://www.klipper3d.org/ (cit. on p. 43).

[77] *Welcome to Mainsail | Mainsail*. Accessed 25-09-2024. URL: https://docs.mainsail.xyz/ (cit. on p. 55).

[78] Erik Westphal and Hermann Seitz. "A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks". In: *Additive Manufacturing* 41 (2021). ISSN: 22148604. DOI: 10.1016/j.addma.2021.101965 (cit. on p. 31).

[79] *What is 3D printing? How does a 3D printer work? Learn 3D printing*. Accessed 20-09-2024. URL: https://3dprinting.com/what-is-3d-printing/ (cit. on p. 13).

[80] *What is a Neural Network? | IBM*. Accessed 22-09-2024. URL: https://www.ibm.com/topics/neural-networks (cit. on pp. 20, 22–24).

[81] *What Is an STL File? – The STL Format Simply Explained | All3DP*. Accessed 17-09-2024. URL: https://all3dp.com/1/stl-file-format-3d-printing/ (cit. on p. 10).

[82] *What is Marlin? | Marlin Firmware*. Accessed 25-09-2024. URL: https://marlinfw.org/docs/basics/introduction.html (cit. on pp. 42, 43).

[83] *What is SLA printing? The original resin 3D print method | Protolabs Network*. Accessed 20-09-2024. URL: https://www.hubs.com/knowledge-base/what-is-sla-3d-printing/ (cit. on p. 11).

[84] Xun Xu et al. "Industry 4.0 and Industry 5.0—Inception, conception and perception". In: *Journal of Manufacturing Systems* 61 (2021). ISSN: 02786125. DOI: 10.1016/j.jmsy.2021.10.006 (cit. on pp. 5, 6, 8).

[85] Sinan Yilmaz. "Comprehensive analysis of 3D printed PA6.6 and fiber-reinforced variants: Revealing mechanical properties and adhesive wear behavior". In: *Polymer Composites* (2023-01). ISSN: 15480569. DOI: 10.1002/pc.27865 (cit. on p. 9).

[86] Laurent Younes. *Introduction to Machine Learning*. 2024. arXiv: 2409.02668 [stat.ML]. URL: https://arxiv.org/abs/2409.02668 (cit. on p. 24).