

Scalable and Generalizable AI Solutions - Applied to Vision-Based Quality Monitoring for Directed Energy Deposition

by

Gijs Johannes Jozef van Houtum

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Mechanical and Mechatronics Engineering

Waterloo, Ontario, Canada, 2024

© Gijs Johannes Jozef van Houtum 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner: Denis Cormier
Professor, Dept. of Industrial and Systems Engineering,
Rochester Institute of Technology

Supervisor: Mihaela Vlasea
Associate Professor, Dept. of Mechanical and Mechatronics Engineering,
University of Waterloo

Internal Member: Amir Khajepour
Professor, Dept. of Mechanical and Mechatronics Engineering,
University of Waterloo

Internal Member: Adrian Gerlich
Professor, Dept. of Mechanical and Mechatronics Engineering,
University of Waterloo

Internal-External Member: John Zelek
Associate Professor, Dept. of Systems Design Engineering,
University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Chapters 3, 4, and 5 consist of articles co-authored by myself and my supervisor, Prof. Vlasea. I conducted the primary research and development for the studies presented in these chapters. This included conceptualizing the research questions, designing the algorithms, and developing the methodologies. I independently designed and implemented the algorithms used in the research. This involved programming and refining the algorithms to address the specific research goals. I was responsible for designing the experiments, including selecting appropriate methods, defining variables, and setting up the experimental protocols. I managed the entire data processing pipeline, which included cleaning, organizing, and annotating the data. This also involved developing procedures for data management and ensuring data integrity. I conducted simulations and performed all data analysis. This included applying statistical methods, interpreting results, and integrating findings into the research. I wrote the drafts of the articles which involved drafting the text, preparing figures and tables, and ensuring that the research findings were clearly presented. Prof. Vlasea provided essential guidance throughout the research process, and offered feedback on the research design, methods, and results. Both authors contributed to reviewing and editing the drafts of the articles, ensuring they met academic standards and contributed to the overall quality of the work.

In summary, while I conducted and managed the research and writing, Prof. Vlasea played a crucial role in providing guidance, reviewing the work, and assisting with editing.

Abstract

This dissertation focuses on enhancing the scalability and generalizability of machine learning (ML) methods for predicting process outcomes in industrial automation, with a particular emphasis on additive manufacturing (AM) via directed energy deposition (DED). The research introduces several novel frameworks to address key challenges in this domain, including data annotation, domain generalization, and real-time adaptability.

First, a novel active learning method is proposed to reduce the need for extensive data annotation. This method optimizes model performance by balancing uncertainty with random sampling, allowing for a transition from exploration to exploitation throughout the learning process. It achieves a notable reduction in annotation requirements, and is designed to be broadly applicable to any ML task involving prediction uncertainty.

Second, the dissertation presents a new validation framework to address domain generalization issues in AM. Traditional ML models often rely on limited and non-diversified datasets, which can lead to an overestimation of performance under real-world conditions. This research introduces a diverse DED dataset designed to simulate various environmental changes typically encountered in real-world scenarios. The framework employs a categorization and validation simulation protocol to assess model performance across different environments. Evaluation of various ML architectures on this dataset provides insights into cross-environment performance and inference latency.

Finally, a training-free domain adaptation framework is developed to tackle real-time process measurement challenges in AM. This framework adapts to diverse operational contexts with minimal annotations through human-in-the-loop interactive annotation guided by the architecture itself. It demonstrates exceptional real-time processing capabilities, and segmentation performance of melt-pool related signatures.

Together, these contributions advance the industrial adoption of AM and enhance its competitive edge over conventional manufacturing processes by improving the scalability, generalizability, and real-time adaptability of ML methods.

Acknowledgements

I am deeply grateful to my research supervisor, Prof. Mihaela Vlasea, for her exceptional mentorship, unwavering support, and invaluable encouragement. Her guidance has been crucial to the success of this research, and working with her has been a true privilege.

I also wish to acknowledge my thesis examining committee members: Prof. Denis Cormier from the Rochester Institute of Technology, Prof. Amir Khajepour, Prof. Adrian Gerlich, and Prof. John Zelek from the University of Waterloo. Their careful review, insightful feedback, and constructive criticism have greatly contributed to the refinement of this work.

My thanks go to the members of the Multi-Scale Additive Manufacturing Laboratory for their willingness to share their expertise and knowledge. Their collaborative support has been highly valuable throughout this research.

I would like to recognize the support of the sponsors of this dissertation, including the Natural Sciences and Engineering Research Council (NSERC) for the Holistic Innovation in Additive Manufacturing (HI-AM) grant NETGP 494158 - 16. I also appreciate the contributions of Xiris Automation Inc. for providing Directed Energy Deposition (DED) datasets and the Federal Economic Development Agency for Southern Ontario (FedDev Ontario grant #814654) for their financial backing.

Lastly, I am grateful to my parents, sister, and significant other for their constant love, support, and encouragement. Their unwavering support has been a source of motivation throughout this journey.

Table of Contents

List of Figures	xi
List of Tables	xiii
1 Introduction	1
1.1 Motivation	1
1.2 Objectives	2
1.3 Outline	3
2 Literature review	5
2.1 Smart manufacturing	5
2.1.1 Additive manufacturing processes	5
2.1.2 Metal additive manufacturing	7
2.1.3 Advances and challenges	9
2.2 Directed energy deposition	9
2.2.1 Process parameters	9
2.2.2 Defects	10
2.2.3 Modelling and control	11
2.3 Quality assessment	13
2.3.1 Active sensors	13
2.3.2 Passive sensors	15

2.4	Data-related challenges	16
2.4.1	Dataset quality	17
2.4.2	Generalizability	18
2.4.3	Redundancy	18
3	Active learning via adaptive weighted uncertainty sampling applied to additive manufacturing	21
3.1	Preface	21
3.2	Introduction	22
3.3	Related work	24
3.3.1	Active learning	24
3.3.2	Vision-based process quality prediction in AM (DED)	26
3.4	Methods	27
3.4.1	Active learning methodology and performance evaluation	27
3.4.2	Adaptive weighted uncertainty sampling	30
3.4.3	DED process quality classification based on melt-pool visibility	32
3.4.4	Experimental setting	36
3.5	Results and Discussion	38
3.5.1	DED feature extraction and classification results	38
3.5.2	Active learning results on reference dataset	39
3.5.3	Active learning results on OpenML and DED datasets	41
3.6	Conclusions	47
4	The impact of domain shift on deep learning model performance in directed energy deposition process quality prediction	48
4.1	Preface	48
4.2	Introduction	49
4.3	Methods	52
4.3.1	DED process quality prediction	52

4.3.2	Dataset	56
4.3.3	Multi-domain validation	57
4.3.4	Experimental details	61
4.4	Results	63
4.4.1	Overall results	63
4.4.2	Relative performance	65
4.4.3	Performance/Speed trade-off	66
4.5	Discussion	68
4.6	Conclusions	69
5	AMVOS: Semi-supervised video object segmentation for additive manufacturing	70
5.1	Preface	70
5.2	Introduction	71
5.3	Methods	74
5.3.1	General VOS description	74
5.3.2	VOS requirements for AM	76
5.3.3	AMVOS: VOS specifically for AM	77
5.3.4	Operation modes	78
5.3.5	Datasets	79
5.3.6	Training experiments	81
5.3.7	Validation experiments	82
5.3.8	Implementation details	83
5.4	Results	83
5.4.1	Running time analysis	83
5.4.2	Semi-supervised performance	85
5.4.3	Interactive performance	86
5.5	Discussion	88

5.5.1	Running time comparison	88
5.5.2	Memory policy performance influence	88
5.6	Conclusions	90
6	Conclusions and future work	91
6.1	Conclusions	91
6.2	Future work	92
	References	94
	APPENDICES	117
A	Annotation protocol	118
B	Active learning	120
C	Domain shift	124
D	AMVOS	127

List of Figures

2.1	Data driven intelligence in smart manufacturing	6
2.2	DED and PBF process comparison	8
2.3	DED processes overview	10
2.4	DED process parameter overview	11
2.5	DED geometrical errors	12
2.6	ML model accuracy versus dataset size.	19
3.1	Active versus passive learning	25
3.2	Query strategy performance evaluation strategy	29
3.3	Instance sampling probability	33
3.4	Recurring objects in DED datasets	34
3.5	DED image feature extraction	36
3.6	DED image examples	37
3.7	DED classification performance	40
3.8	Active learning performance on the "Horizon" dataset	42
3.9	Active learning performance on all datasets	44
3.10	Active learning performance by dataset source	45
3.11	Active learning performance by batch size	46
4.1	DED video recording image examples	54
4.2	Validation dataset differences and examples	59

4.3	Overall classification results	64
4.4	Relative performance between architectures and validation datasets	67
5.1	AMVOS-A; Semi-supervised video object segmentation	75
5.2	AMVOS-S; Object specific video object segmentation	79
5.3	AMVOS-I; Interactive video object segmentation	80
5.4	AMVOS; run-time analysis	84
5.5	AMVOS; semi-supervised performance analysis	86
5.6	AMVOS-I; interactive performance analysis	87
5.7	Annotation and prediction disagreement	89
A.1	Annotation protocol used to create DED datasets described in Chapters 3, 4, and 5.	119
C.1	DED scene examples	124
C.2	Classification accuracy of 83 architectures	125
C.3	Inference speed for 83 architectures	126
D.1	AMVOS architecture blocks	128
D.2	AMVOS architecture blocks	129

List of Tables

3.1	Adaptive weighted uncertainty sampling algorithm	32
3.2	DED image classification	34
4.1	Xiris dataset statistics	58
5.1	AMVOS training and validation datasets	80
5.2	VOS architectures comparison	88
B.1	Pool-based active learning algorithm	120
B.2	DED classification performance on all datasets	121
B.3	Active learning simulation procedure	122
B.4	Active learning performance	123

Chapter 1

Introduction

1.1 Motivation

Additive manufacturing (AM), particularly Directed Energy Deposition (DED), is a transformative technology in smart manufacturing. It enhances the product supply chain from design to final component, offering significant benefits in customization and material efficiency. AM reduces operational footprints, improves scalability, and supports on-demand production [1].

A key challenge in AM process modeling is the prediction of defects. Current models often fail to anticipate local geometry variations caused by stochastic events, as well as the occurrence, location, and size of defects. This makes it more challenging to achieve consistent fabrication of full-density parts with predefined mechanical properties and dimensional accuracy compared to traditional manufacturing methods [2].

Recent advancements in sensor technology and data acquisition have dramatically increased the volume of data available before, during, and after part fabrication. This influx of data presents new opportunities for discovering insights and enhancing quality throughout AM processes. Efforts to improve process repeatability and part quality now increasingly focus on comprehensive monitoring and inspection [3].

Machine learning (ML) has become a critical tool in AM, utilizing abundant data to predict process quality and detect defects. However, current ML approaches often emphasize optimizing model performance over improving scalability and generalizability to new, unseen environments. This limitation arises from the lack of diversity in training and testing datasets. Developing effective supervised ML models requires extensive and high-quality data annotation, which is both time-consuming and costly. Additionally, creating

diverse datasets remains challenging as existing practices typically focus on specific AM machines and conditions [4, 5, 6].

To facilitate broader industrial adoption of AM, it is crucial to enhance the efficiency and scalability of ML applications. This research aims to address these needs by developing ML solutions that not only generalize well to new environments but also scale effectively [7].

1.2 Objectives

The primary goal of this research is to develop ML tools for AM, specifically DED, that enhance both the generalizability and scalability of ML models in novel operating environments. To achieve this goal, the following sub-goals are proposed:

- Generalizability: Develop ML solutions that are not restricted to AM or DED but are designed to be applicable across a broad range of industrial applications. This will ensure that the models perform effectively in diverse environments beyond those they were specifically trained on.
- Scalability: Create ML solutions that reduce the need for extensive training, large datasets, and prolonged processing times. This will promote the efficient adoption and deployment of the models on a larger scale within industrial settings.

To address these goals, the following concrete objectives were developed:

- Annotation reduction: Investigate and develop new ML algorithms that reduce the required amount of annotated data to develop high performing ML models, and demonstrate them on DED. Minimizing annotation decreases model development time, leading to improved scalability.
- Domain generalization: Investigate the impact of different experimental operating conditions, on ML performance in AM, and specifically DED. An understanding of the effect of ML model deployment on performance in un-seen environments provides insight into the generalizability.
- Semi-supervision: Investigate and develop novel semi-supervised ML solutions that are able to adapt to new un-seen operating environments using less annotated data

and without the need for model re-training. Adaptation guided by a few annotated data instances, instead of re-training, offers generalizability and scalability as annotation and model development time are reduced.

The algorithmic contributions of this research, designed to address these objectives, are applied to DED datasets. The DED datasets used in this thesis are created using the annotation protocol as described in Figure A.1 which can be found in Supplementary material A. These algorithms are developed with generalization in mind, ensuring they can be adapted for application in other domains beyond the specific focus of this thesis. Besides these main objectives, DED specific objectives were developed:

- Measurement quality: Investigate and develop new ML algorithms that are able to predict the quality of DED in-situ process measurements for further information extraction. Faulty measurements could potentially disrupt secondary algorithms using the measurements, such as closed-loop control algorithms, decreasing process repeatability and part quality.
- Melt-pool segmentation: Create ML-based DED melt-pool segmentation algorithms able to operate in real-time and easily adaptable to new DED environments. As such, cross-environment real-time closed-loop melt-pool geometry control could be enabled to further improve repeatability and part quality. Melt-pool is defined in this thesis as any observation related to the physical molten material created by the DED process.

1.3 Outline

This thesis is organized into six chapters, with the current chapter serving as the introduction. Contributions are provided by Chapters 3, 4, and 5.

Chapter 2 provides background information and a literature review to support the understanding of the material in subsequent chapters. It begins with an introduction to AM and its role in the fourth industrial revolution, followed by a discussion of DED processes and common quality monitoring methods. At last, ML and data-related challenges in DED are discussed.

Chapter 3 presents a novel and patented active learning algorithm, which is applicable to many datasets and ML tasks and significantly reduces the required annotation for ML

model development. Additionally, this chapter develops and evaluates a new ML vision-based classification architecture for predicting DED process quality.

Chapter 4 explores the domain generalization capabilities of ML model architectures with a focus on AM and DED. It studies the impact of various newly defined changes in DED operating environments on quality classification across multiple ML architectures, offering insights into the applicability of ML models beyond their training data and highlighting the limitations of conventional ML architectures.

Chapter 5 introduces a novel training-free adaptive ML segmentation framework designed for general applicability to any segmentation task and object, with a specific focus on AM and DED. This framework adapts to different environments without excessive annotation and functions as an active learning recommendation system, as demonstrated through DED melt-pool segmentation in various operating environments.

Chapter 6 discusses the conclusions drawn from the research and outlines directions for future work.

Chapter 2

Literature review

2.1 Smart manufacturing

Smart manufacturing is a central element of the fourth industrial revolution - also named Industry 4.0 - focused on increasing the quality, productivity, sustainability, and flexibility of the production process by leveraging digital technologies such as the Internet of Things, cloud services, big data and associated data analytics [7].

Artificial intelligence (AI) is a key digital technology in smart manufacturing and AM, which represents a group of computer techniques focused on the extraction of knowledge from structured and unstructured data sources - such as sensor data acquired from process measurement - and making appropriate decisions to improve process operation. ML, and specifically deep learning methods within the AI domain, demonstrate outstanding performance in smart manufacturing applications by enabling automatic feature learning and high GPU-based data processing capabilities in the big data era [8] (Figure 2.1).

2.1.1 Additive manufacturing processes

According to the ISO/ASTM Standard 52900, object creation through manufacturing can be achieved by formative, subtractive, or additive shaping. While the former two methods achieve the desired object shape through the application of pressure or the selective removal of material respectively, the latter refers to processes which create objects using successive addition or joining of material. The properties of AM-produced parts are determined by

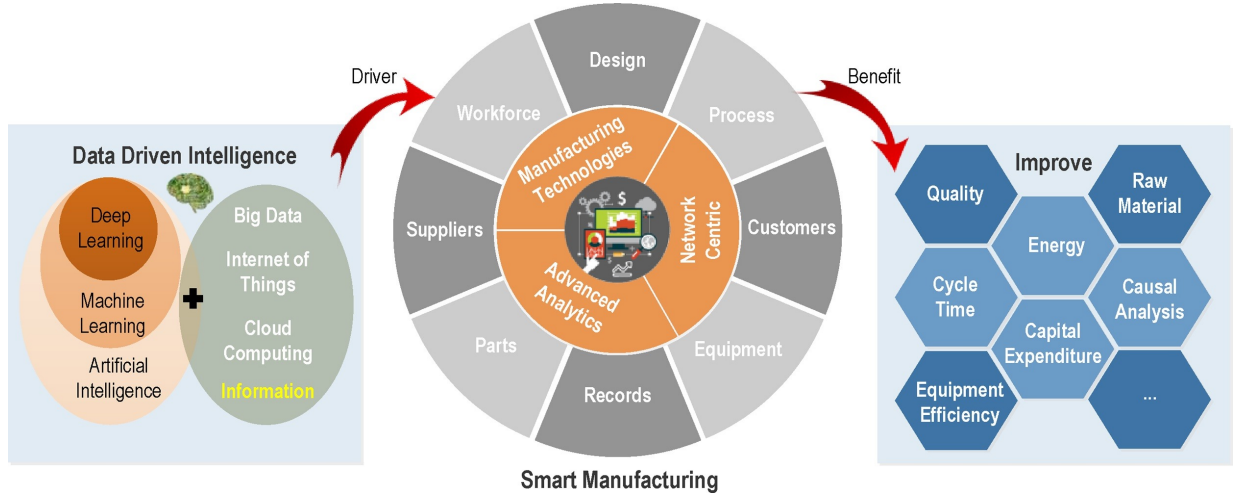


Figure 2.1: The role of data driven intelligence in smart manufacturing as shown in [8].

the type of material (polymer, metal, ceramic, or composite), principle of fusion or bonding, the added material (feed-stock), and the machine architecture. The fundamental classes of AM technologies can be categorized into binder-jetting, DED, material extrusion, material jetting, powder bed fusion (PBF), sheet lamination, and vat photo-polymerization [9].

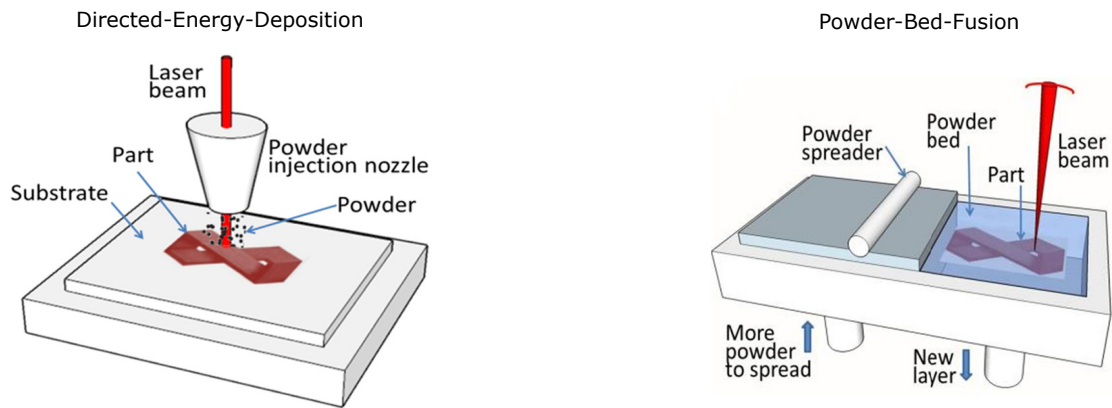
Single-step metal-based AM processes can be classified based on the material feed-stock and/or energy source and generally fall into two categories, DED and PBF, with popular uptake within the aerospace, medical, and automotive industries [10]. PBF is a process in which thin layers of powder are spread onto a build plate, then an energy source selectively fuses regions of the powder bed. Depending on the energy source, the process takes place in an inert atmosphere (laser-based systems) or in a vacuum (electron beam systems). When one layer is fused, a new layer of powder is applied, and the process is repeated until a three-dimensional (3D) part is obtained. DED is a process where a material (powder or wire) is delivered to the substrate using a nozzle assembly, with an energy source (laser, electron beam, plasma arc, gas metal arc) interacting with the incoming material and the substrate to melt and solidify a sequence of material tracks. Both processes require a material substrate (or base plate) to anchor the parts onto and require support structures to be manufactured for overhang features.

2.1.2 Metal additive manufacturing

When considering PBF and DED, there are important differences to consider. For instance, when comparing laser-based PBF and DED technologies which employ powders, the presence of powder in a powder bed when compared to a powder delivery stream creates different melt-pool dynamics and heat transfer phenomena [11, 12]. In addition, from a temporal resolution perspective, the laser beam velocities used in PBF are significantly higher than DED, leading to a higher occurrence of liquid metal instability due to Marangoni convection and liquid metal spatter effects [13]. Furthermore, the spatial resolution is much more refined in PBF, where the lower beam spot diameters, typically 50-100 μm , have lower vaporization thresholds than in DED, where the laser beam spot diameters are in the scale of 500 μm - 2 mm [14]. A visualization of these processes, and their process parameters, is provided in Figure 2.2 [15].

Metal AM, and particularly DED and PBF, provide advantages compared to formative and subtractive manufacturing technologies. The additive method of part production makes it possible to produce highly complex geometries without creating excessive waste material. The design potentials in metal AM are also enabling the consolidation of assemblies, which reduces complexity and the likelihood of failure by virtue of assembly and joining operations [16]. Another advantage of AM is mass customization, as the production of product families with custom features is possible at a low cost, without the need for drastic changes in tooling or the production line workflow [17]. Furthermore, the product supply chain is greatly simplified by AM, as digital 3D part designs can be directly produced by AM machinery, enabling on-demand manufacturing from raw material to part. This also offers the potential to manufacture in atypical, remote environments - such as space [18]. These advantages make AM a valuable asset in the transition to smart manufacturing as discussed in Section 2.1.

The application domain of metal AM processes differs based on the capabilities of each class of technologies. A detailed comparison between DED and PBF processes capabilities is provided in Figure 2.2, showing that PBF processes generally provide a smaller build chamber size, therefore limiting the size of the parts being manufactured, compared to DED processes. Although the layer deposition rate of PBF is much higher compared to DED, the production time is generally longer due to the smaller layer thickness. The dimensional accuracy of PBF systems is, however, much higher, making the incorporation of intricate features possible. Post-processing is therefore less demanding for PBF processes. Wire-DED processes using an electric arc as an energy source - which is very similar to welding - provide the fastest but crudest way of manufacturing large parts while laser-PBF provides slow but accurate production of smaller parts. DED is therefore commonly used for the



Comparison

Process	DED			PBF	
	Powder	Wire		Powder	
Heat source	Laser	E-beam	Electric arc	Laser	E-beam
Nomenclature	DED-L	DED-EB	DED-PA/DED-GMA	PBF-L	PBF-EB
Power (W)	100–3000	500–2000	1000–3000	50–1000	
Speed (mm/s)	5–20	1–10	5–15	10–1000	
Max. feed rate (g/s)	0.1–1.0	0.1–2.0	0.2–2.8	–	
Max. build size (mm × mm × mm)	2000 × 1500 × 750	2000 × 1500 × 750	5000 × 3000 × 1000	500 × 280 × 320	
Production time	High	Medium	Low	High	
Dimensional accuracy (mm)	0.5–1.0	1.0–1.5	Intricate features are not possible	0.04–0.2	
Surface roughness (μm)	4–10	8–15	Needs machining	7–20	
Post processing	HIP and surface grinding are seldom required	Surface grinding and machining is required to achieve better finish	Machining is essential to produce final parts	HIP is rarely required to reduce porosity	

Figure 2.2: A comparison of DED and PBF processes as shown in [15].

production of large components with low geometric complexity and also for repairing larger components in the automotive and aerospace industry [19].

2.1.3 Advances and challenges

AM is experiencing rapid growth, driven by its advantages over traditional manufacturing methods. It is projected to expand at a compound annual growth rate (CAGR) of 23.6% through 2030. The current global AM market is divided into 53% for tooling, 37% for manufacturing functional parts, and 10% for prototyping pre-manufactured parts [20].

To advance industrialization and broader adoption, ongoing research is crucial to enhance the performance of metal AM processes. Challenges persist in achieving consistent fabrication of full-density parts with predefined mechanical properties and dimensional accuracy using current technologies [2]. These challenges arise because both the shape and material properties of a part are developed during the AM process. The creation of parts involves complex interactions among design, materials, and processes across five major stages, often executed iteratively: designing, process planning, building, post-processing, and testing [4].

2.2 Directed energy deposition

This thesis primarily focuses on DED processes, which are detailed in this section. DED involves depositing metal wire and/or powder materials directly onto a workpiece through irradiation with a laser beam, electron beam, or electric arc. During operation, the deposition head—comprising the material delivery nozzle, laser beam, and associated optics—moves relative to the substrate, creating a molten metal pool, or melt pool, at the interaction point. Sequential movement along a predefined pattern within each layer, combined with successive layering, forms 3D geometries. Although similar to welding, which is used for joining or repairing metal parts, DED is specifically designed for additive manufacturing [21] (see Figure 2.3).

2.2.1 Process parameters

Process parameters significantly impact the operating conditions of a DED system and the quality of the produced parts. These parameters can be classified into three categories

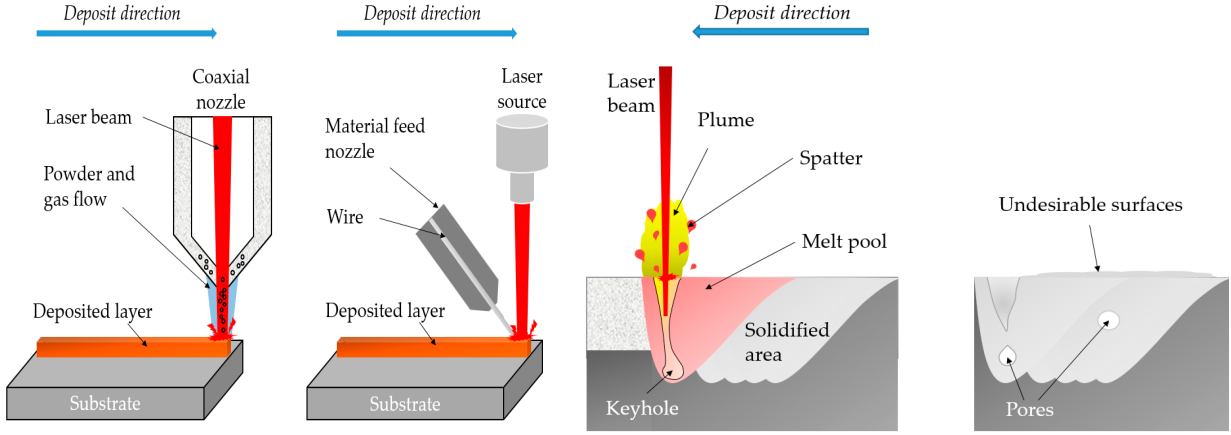


Figure 2.3: Left-to-right; Powder-DED, Wire-DED, cross section view of melt-pool and corresponding keyhole leading to porosity from [21]

based on recent reviews: (i) fixed, (ii) pre-defined, and (iii) controllable parameters (see Figure 2.4). Fixed parameters are specific to the DED machine and its components, such as the energy source, feed-stock system, and shielding gas systems. These parameters cannot be adjusted or are difficult to modify after the machine has been fabricated. Pre-defined parameters remain constant throughout the entire build of a part but can be adjusted between different builds. Controllable parameters, on the other hand, can be set to a fixed value, adjusted to different values during the build, or actively controlled while the process is ongoing.

2.2.2 Defects

Improper configuration of process parameters can lead to various defects in manufactured parts, categorized into micro-structural, geometrical, and mechanical issues.

Micro-structural defects include porous anomalies such as keyhole phenomena [28], entrapped gas pores, and lack-of-fusion defects. Keyhole gas pores form when power density exceeds a material-specific threshold, influenced by recoil pressure and Marangoni convection [29, 30] (see Figure 2.3). The interactions between laser and material induce turbulence in the melt pool, leading to gas porosity and potential spatter [31]. Predictive models based on normalized enthalpy, considering factors such as energy source power and scan velocity, can estimate porosity [29, 32]. Entrapped gas pores may also result from inconsistencies in powder delivery or variations in gas solubility [33]. Lack-of-fusion

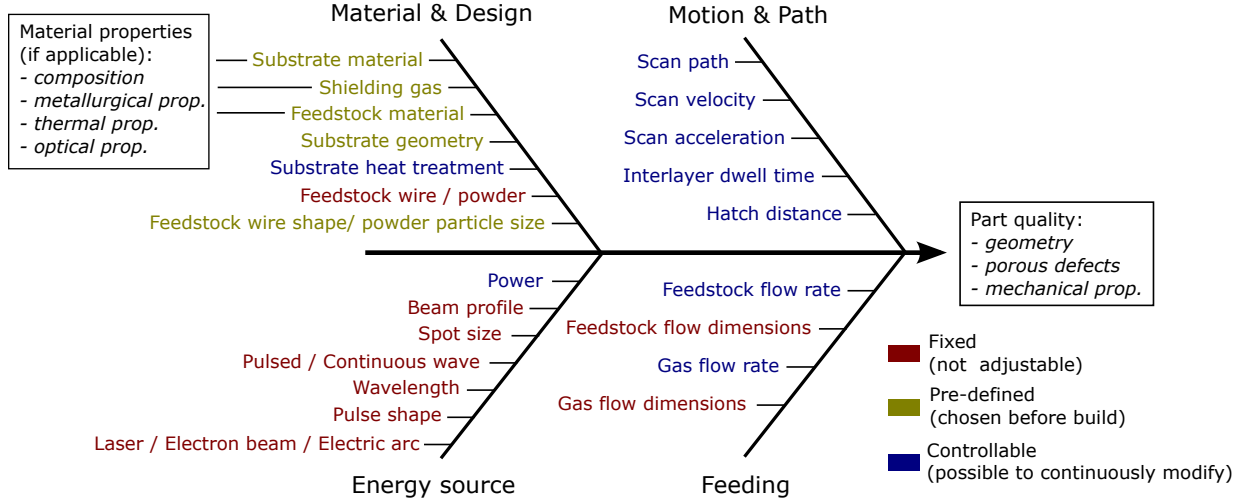


Figure 2.4: Process parameters from [15, 22, 23, 21, 24, 25, 26, 27] presented in fish-bone diagram according to [26, 27] and labeled based on their adjustability as proposed by [3]

pores arise when insufficient energy fails to bond layers or scan lines, influenced by factors like layer thickness, energy source power, and scan velocity [23].

Geometrical inaccuracies often arise from variations in melt-pool dimensions, cracking, humping, spatter, and kinetic recoil [34, 35]. The Marangoni effect can cause local surface tension variations, impacting melt-pool geometry [22]. Cracks may form due to internal residual stresses from sequential heating [36]. Spatter, or particle ejection, results from high-power localized boiling, causing inaccuracies when particles land on the substrate [37]. Kinetic recoil involves powder rebounding and landing in unintended locations [35]. Humping, the rise of material above the melting plane due to surface tension, can also cause geometric anomalies [38]. Anomalies in one layer can propagate, affecting the consistency of the powder stream and laser beam, leading to significant deviations (Figure 2.5).

Mechanical properties, such as tensile strength, hardness, and fatigue, can be influenced by process parameters affecting the micro-structure [39, 40, 41]. Residual stresses arise from steep thermal gradients between the energy source and the surrounding material [22].

2.2.3 Modelling and control

Analytical and numerical modeling methods are crucial in DED research for predicting product quality based on process parameters. However, modeling the complex interactions

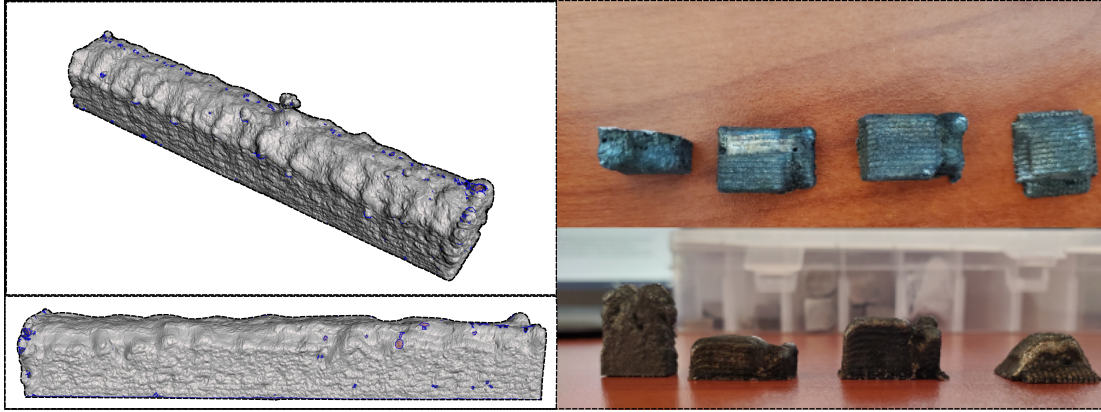


Figure 2.5: Geometrical errors in two different parts, scanned using a AICON SmartScan structured light optical scanner.

between lasers and materials, along with numerous process parameters, presents significant challenges. A review of recent papers on various physics-based models [42, 43, 44, 45, 46, 47, 48, 49, 50] reveals several key observations.

Many models fail to account for local imperfections in materials (such as surface roughness and micro-structure), environmental conditions (such as air temperature and vibrations), and equipment factors (such as tolerances and non-steady-state behavior). This oversight leads to localized prediction errors. Accumulated errors can cause significant deviations in geometric accuracy, porosity, and micro-structure from steady-state assumptions by the end of the build process (see Figure 2.5; these aspects are seldom addressed in modeling efforts. A major challenge for current models is their inability to accurately predict defects. Existing approaches struggle with forecasting local geometry variations due to stochastic events and anticipating defects, including their occurrence, position, and size.

In-situ monitoring and control of the AM build process could significantly enhance the reliability, repeatability, and accuracy of manufactured parts through feedback mechanisms [51]. Most commercial metal AM processes are however open-loop or controlled in a feed-forward manner without feedback, leading to micro-structural, mechanical, and geometric errors (see Section 2.2.2). By detecting and correcting defects through process parameter adjustments during the printing process, part quality and repeatability could be improved [5]. However, implementing such systems requires the capability to adjust process parameters in real-time and to observe and measure the quantities being controlled.

2.3 Quality assessment

Advancements in sensor technologies and data acquisition methods have led to a drastic increase in AM data available before, during, and after part fabrication [3]. The different types of data generated throughout the design-to-product cycle are creating new opportunities for knowledge discovery and quality improvement throughout AM processes.

The DED process consists of many interconnected and coupled physical events with several possible paths for momentum and energy transfer. Some of these physical events in order of appearance include: laser delivery, particle/powder delivery, laser/powder/shielding gas interactions, melt pool initiation, melt pool energy/stability/morphology, heat loss (radiation, convection, conduction) and solidification [52]. The state of these phenomena directly affects process quality, by influencing defect creation. Measuring the state of these phenomena through physical measurement of the melt-pool dynamics and geometry, micro-structural defects or mechanical properties is challenging during the manufacturing process, as process interruption or part-destruction is necessary for the measurement. Object testing, with the use of active sensors, or in-situ process measurement using passive sensors is therefore preferred as feedback sensors provide non-destructive testing/measurement methods [53].

2.3.1 Active sensors

Active sensors test the object and measure its response, without damaging the object; such sensors are also called non-destructive. The most common methods used in DED include radiography (X-ray), ultrasonic testing, eddy current testing and structured light [54, 55].

In X-ray computed tomography (CT) an X-ray beam, typically with a wavelength between 0.01 and 10 nanometers [56], passes through the object of interest and a scintillator is used to convert the X-ray beam to visible light which is captured by a vision camera [57]. CT is perhaps the most accurate and reliable defect detection method as morphology, size, frequency and location of the defect can be extracted from CT data [55, 58]. Furthermore, high temporal resolution can be achieved to observe and measure the internal melt-pool flow dynamics as showed by Aucott et al. [59] who were able to measure the melt-pool geometry, and the size, direction, position and velocity of particles inside and outside the melt-pool. The maximum particle velocity measured was 0.54 m/s using a 0.0508 nanometer X-ray beam, by a 800 by 600 (8 bit) pixel resolution vision camera with a frame rate of 1000 frames per second. The resulting rate of data acquisition is therefore very high at 480

megabytes per second. The authors of [57] showed that particle velocities also can be approximated using a lower frame rate of 20 frames per second using a camera resolution of 1024 by 1024 (8 bit) 1.97 micrometer pixels, leading to a lower data acquisition rate of approximately 21 megabytes per second. The cost of CT equipment is however high, and the related ionizing radiation requires shielding for personal and other equipment [60]. Furthermore, if in-process quality detection is pursued, image-based defect detection algorithms have to be developed capable of running faster than the image acquisition rate. Such an approach, while scientifically valuable, is not economically viable to implement in commercial systems.

Ultrasonic testing is another active sensor method where a probe is used to generate sound waves, normally with center frequencies ranging from 0.1-15 MHz ($3^{12} - 2^{10}$ nanometer), in the object and a sensor is used to measure the resulting wave after travelling through the medium. Differences between the in- and output are used to determine whether there are defects present in the material [61, 62, 63]. Typical sampling frequencies used for measurement of the signal is 50 [64] or 100 [65] Mega Hertz, resulting in 50000-100000 data instances per second. Solidified defect detection in the form of porosity or surface defects is where ultrasonic is commonly used for. This form of testing is of relatively low cost compared to CT, but the generated information density is much lower. Furthermore, the detection of melt-pool geometry or flow dynamics is impossible, as the measurement takes place after solidification.

Eddy current testing is, in its basic form, a coil of conductive wire that is excited with an alternating electrical current, or probe. This probe produces an oscillating magnetic field at the same frequency as the current running through the coil. When the probe approaches a conductive material, currents opposite to the ones in the coil, called eddy currents, are induced in the material [66]. Variations in the electrical conductivity, magnetic permeability and the presence of defects cause a phase and amplitude that can be detected by measuring the impedance changes in the coil [67]. The conventional eddy current techniques are capable of detecting cracks and porosity [68, 69] and the data produced is a sampled scalar signal in time much like ultrasonic testing.

Structured light refers to methods where light in the form of a line or pattern is projected onto the object and measured by a vision camera. If calibrated, the relative position and angle between the light source and the camera are known and allow for the measurement of the objects geometry. Main use cases are the geometrical measurement of the solidified melt-pool just after solidification [70, 71, 72, 73]. Data acquisition is highly dependent on the resolution and frame rate of the camera allowing for low and high-speed implementations equivalent to the previously discussed X-ray imaging implementations.

2.3.2 Passive sensors

Passive sensors measure signals generated by the DED process itself, instead of actively generating them. The interaction between the energy source and materials that occurs during DED, emit energy in a variety of forms. Optical emission in the form of ultraviolet (UV), visible (VIS) and infrared (IR) are the most dominant types of electromagnetic radiation produced by the process [74]. Acoustic emissions of DED, can be divided into air-borne and structure-borne emissions that originated from the stress waves induced by changes in the internal structure of a work piece, the surface the metal vapour and laser beam back reflection [75]. Additionally, the electric charge generated inside energy-source induced plasma is the third signal that can be used. One end of the circuit is linked with the base material, while the other end is connected with the energy source and a voltage measurement between them can be used for sensing purposes [66].

Contact type acoustic emission sensors mainly monitor the stress waves generated by high temperature and high pressure inside the equipment [76] or substrate [77, 78] it is attached to and usually detects wavelength less than 200 kHz. Porosity, cracks and internal stresses are normally monitored using this technology. The non-contact type of sound signal detection mainly monitor the pressure waves when plasma and metallic vapour occur in the range of 20 Hz–20 kHz and could be used to relate plasma or vapour changes to the creation of defects [74, 79].

Photo-diodes, also called pyrometers, are devices used to detect process radiation of in the ultraviolet 200-400 nanometer (UV), visible 400-700 nanometer (VIS) or infrared 700-1700 nanometer (IR), as detection wavelengths commonly range between 260 and 1700 nanometer [80, 81, 82]. This device generates a scalar electric signal, which is proportional to the radiation intensity and temperature [83]. The contributions from the plasma plume, spatter, and melt-pool radiation/temperature are typically measured using photo-diodes [82]

Spectrometers are another monitoring method, which is used for similar purposes as photo-diodes to detect the plasma plume, spatter and melt-pool temperature. An array of photo-diodes, which cover different ranges of the UV, VIS and IR spectrum are usually combined to capture a large range of the electromagnetic spectrum in a spectrometer. The electronic processing elements of the spectrometer then translate the electrical signal from each photo-diode into a power associated with the wavelength range [84].

Cameras are generally used to detect process radiation in the UV, VIS and IR ranges similar to photo-diodes but generate array-like signals in the form of images instead of scalar signals. They are typically placed co-axially [85] or at an angle [86, 87] relative to

the energy source to detect the thermal footprint (IR) or geometry (VIS) of the melt-pool, spatter or the plasma plume [74] in space and time. A high frame rate and image size often result in large amounts of data being generated. At 1500 frames per second at 1024 by 1024 pixels (8 bit) per frame resulted in data generation at a rate of 1.5+GB per second in [86]. Wirth et al. [88] used a high speed camera generating images of 256 by 256 pixels (8-bit) at 67,000 frames per second which leads to a the generation of approximately 4.4 GB of data per second which enabled them to observe melt-pool flow dynamics as well. Cameras can provide a rich signal, that is both spatially and temporally resolved.

This overview of sensor data available for DED processes offers a glimpse of the range of options available for capturing process phenomena. The fabrication of components by virtue of fusion of units of materials together, in tandem with sensor data, can generate a vast amount of data. One of the most challenging aspects in this field is extracting valuable information from data.

2.4 Data-related challenges

Perhaps the most impressive and well-known results of AI/ML come from fields where human opponents are directly defeated by, or competing with the AI. ResNet [89] defeated humans in image classification on ImageNet, while AlphaGo [90] defeated human champions in the game of Go. Google recently launched a chatbot called Meena [91] that is able to have specific conversations with humans, while OpenAI developed a large natural language processing engine called GPT-3 [92].

Their impressive performance is however largely made possible due to the use of vast annotated datasets for training purposes of the AI. ResNet used ImageNet, which contains about 1.2 million image-label pairs, while AlphaGo used more than 38 million game-play positions to train, and Meena was trained on 391GB of text data, equivalent to 341,000 books. GPT-3 exceeds the previous training efforts by a large margin, as roughly 2TB of text data was used to train the algorithm. Furthermore, training all 175 billion parameters of GPT-3 on the Microsoft Azure AI supercomputer [93] costed 12 million dollars [94]. This mindset of creating larger models using as much data as possible is caused by the unreasonable effectiveness of data, as proposed in the work by Halevy et al. [95]. One of the authors Peter Norvig, who served as Google’s director of research, stated: ”We don’t have better algorithms. We just have more data”.

2.4.1 Dataset quality

Creating large datasets can however be problematic. In fact, in real-world industrial applications data are hard to acquire and if it exists freely, it is often low quality. A comprehensive survey [96] among 227 data scientists, AI experts and business stakeholders from 5 continents showed that 96% of the companies run into training related problems with data quality and annotation. Furthermore, 72% report that more than 100,000 labeled data items are required for deployment of AI/ML models.

The creation of quality datasets is time-consuming and often difficult [97]. In video annotation, human-like judgements are required on an image-by-image basis, often for multiple objects. Furthermore, segmentation of objects [98] requires even more time, as object boundaries have to be determined on the pixel level [99]. Crowd-sourcing could be a solution, but is expensive and misses domain knowledge, often leading to low quality annotations [100]. To minimize bias, annotation is ideally performed by multiple people [101], who all need to be experts in the field. Domain can be an issue, as you need ideally a group of people trained on the subject and the capability to annotate; which is not trivial. Also, preference is an issue, as not everyone agrees or wants the same thing with equal data. Without proper annotation by a group of experts, the resulting datasets can direct model training towards creating bias, even in very large datasets such as as ImageNet [102]. Access to data is not trivial either. The COVID-19 pandemic can serve as an example, where not enough data was available beforehand to predict infectious behaviour at the global scale [103]. Developing models for rare effects is therefore difficult and prone to data-related issues.

Other studies state that data alone is not enough. The authors of [104] argued that learning algorithms should be improved to be able to perform on data beyond what is presented during training, which requires the usage of domain knowledge. Generalization towards new unseen cases is a big problem, as is pointed out by the authors in the work by Alcorn et al. [105]. They show that popular ML models on ImageNet, such as ResNet and Inception [106], are heavily over-fitting to the ImageNet dataset since slight changes in the poses of the objects lead to incorrect classifications in 97% of their pose space [106]. Diversity in datasets is therefore necessary to train models that generalize well beyond their training data [107]. More data does not necessary lead to better performance as pointed out by Zhu et al. [108]. Google Research [109], came after comprehensive investigation of the ImageNet dataset to the conclusion that 10% of the data is redundant and does not lead to improved model performance when trained upon.

Creation-related challenges are largely referring to supervised machine learning techniques, which require input-output data pairings for training purposes. To address

these data-problems, research has shifted interest into the development of non-supervised methods that provide data-efficient machine learning methods [110].

2.4.2 Generalizability

Creating ML models that are robust and generalize to unseen data is challenging due to data accessibility and creation problems as discussed in the previous Section 2.4.1. With this knowledge in mind, it is important to take another look at the current state of ML in AM, focused on assessing the robustness and generalizing capabilities of the available models instead of performance. Since it is challenging/impossible to have access to, and test all trained ML models in literature on unseen data, or assess their data creation procedure, we focus on the data sources first. The data sources used for training purposes provide insight into the application domain and the generalizing capabilities of the model.

The review papers [4, 5, 6] summarize the state-of-the-art ML applications, from the design, to the build and validation stages of part production in AM. All reviewed papers, which use experimental datasets (73 in total) acquired from process measurement, are analyzed by their data sources leading to the following 2 observations. (1) All studies use a single experimental setup with a specific set of hardware components. It is therefore questionable if these models generalize to other AM machines, sensors within the same class, or any change made to the specific hardware setup used for the generation of the data. (2) A limited range of process operating conditions is used for the generation of the datasets. A change in part design, process parameters or any other change in process operation could therefore lead to decreased model performance.

Although the applications of ML in AM provide insight into how input data (such as sensor measurements) can be used to make predictions using ML, their generalizability to new unseen data is questionable. Furthermore, since the models are not trained and tested on the same datasets, it is impossible to compare performance. Any change made to the data-generating system could therefore require the generation, annotation, and re-training of the model on new data.

2.4.3 Redundancy

Assessment of redundancy in datasets used for training and validation purposes of the ML models developed is challenging without access to the actual datasets. However, a group of ML models developed to perform a similar task by training on datasets which are generated by similar data sources might provide insight into the influence of model choice and dataset

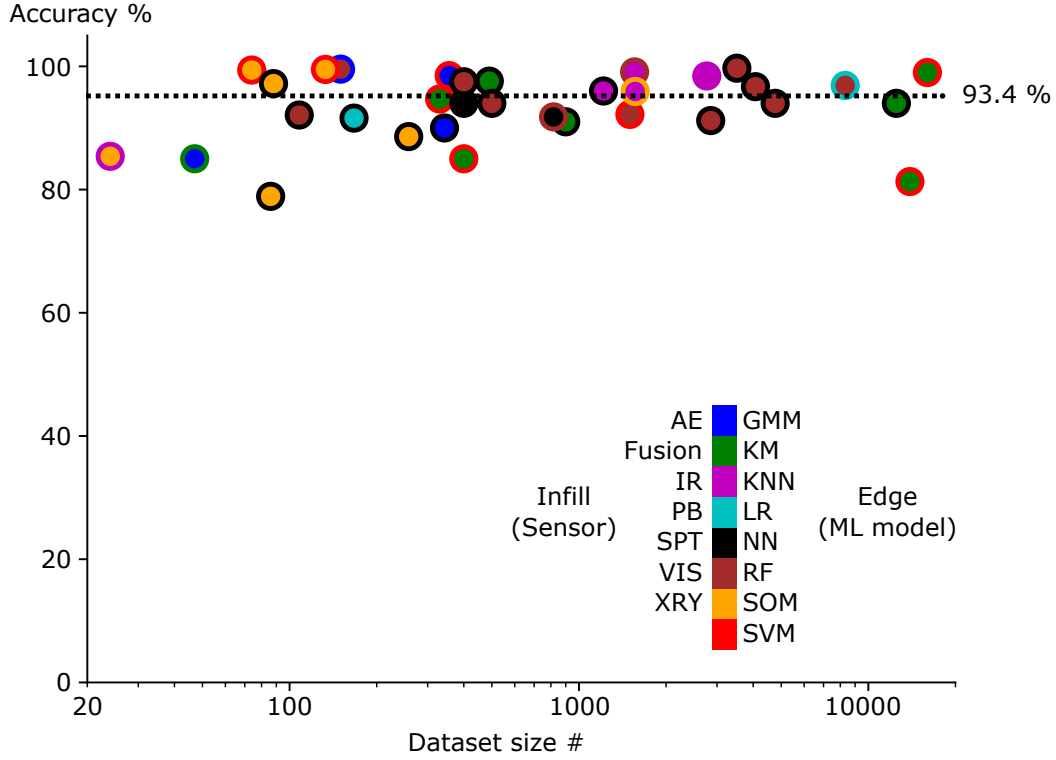


Figure 2.6: ML model accuracy versus dataset size.

size on the model performance. All 34 studies in [111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144] provide state-of-the-art ML models to classify input data from sensor-based sources into defective or non-defective categories.

The sensory sources used in these studies are acoustic emission (AE), infrared/thermal cameras (IR), plasma probes (PB), spectrometers (SPT), visual cameras (VIS), X-ray imaging (XRY) or a combination of the former (Fusion). Measurements from these sensors are used to predict whether a defect, in the form of geometrical surface defects or sub-surface porosity type of defects, has occurred or not. The different ML models used are Gaussian Mixture Models (GMM), K-means clustering (KM), K-nearest neighbors (KNN), Logistic regression (LR), Neural networks (NN), Random forest (RF), Self-organizing maps (SOM) and Support vector machines (SVM).

Figure 2.6 illustrates the highest classification performance achieved for each research paper as a function of dataset size, with an impressive average accuracy of 93.4%.

Although a linear trend was fitted to examine the relationship between dataset size and performance, the resulting slope had a p-value of 0.9, suggesting that dataset size does not significantly impact performance. This indicates that the machine learning models are generally robust and perform well across different dataset types and sizes. Additionally, the consistently high performance across all data sources suggests that each dataset is sufficiently information-rich to predict defects effectively.

However, despite these promising results, there are concerns about the generalizability of the models. The datasets used are relatively small compared to the 100,000 instances recommended by [96], raising questions about the robustness of the models in real-world applications. The observed independence of performance from dataset size may indicate redundancy, as the models perform similarly regardless of dataset size. Thus, while the performance appears high, the implications for model generalizability and practical efficacy should be carefully considered, especially in light of the relatively small dataset sizes used.

Chapter 3

Active learning via adaptive weighted uncertainty sampling applied to additive manufacturing¹

3.1 Preface

The acquisition of large amounts of high-dimensional data is becoming prevalent in additive manufacturing, arising from sensor integration in such processes. Although supervised machine learning has gained popularity to predict process quality from such data, annotation can be time-consuming and labour intensive. Active learning is a sub-field within machine learning concerned with maximizing model performance using the least amount of annotated data. The focus of this study is twofold. Firstly, a novel active learning method is introduced, called adaptive weighted uncertainty sampling (AWUS) which balances uncertainty with random sampling using the model change between active learning iterations. This way, AWUS is able to adapt from exploration of the instance space to exploitation of model knowledge throughout the active learning process. Secondly, a novel feature extraction and classification method is proposed for directed-energy-deposition additive manufacturing processes which predicts image and process quality based upon the visibility of the melt-pool using the visual signature of occluding features in the image. AWUS is compared against 6 state-of-the-art query strategies on 28 datasets from the OpenML database and 8 in-situ machine vision recordings of directed-energy-deposition processes using 4 different classifiers. The results

¹Chapter is adapted from a published article [\[145\]](#)

show that AWUS outperforms the state-of-the-art across the board of the datasets, classifiers and active learning batch sizes. Furthermore, the application of AWUS reduces the amount of necessary annotations by 20 to 70% in 90% of our experiments compared to random sampling.

3.2 Introduction

Machine learning (ML) has been applied to many areas of the additive manufacturing (AM) development cycle, and specifically to directed-energy-deposition (DED) processes. The appearance and geometry of the molten material, or the melt-pool, at the point of interaction between the energy source and material are popular features used for the prediction of defects or geometry control. In-situ imaging is a popular and low-cost solution to observe the melt-pool, with image processing heuristics [146, 147] or ML feature extraction methods [148, 129] being used to extract melt-pool features or predict defects.

Process instability or sub-optimal camera settings are often neglected process quality metrics [4]; however, such metrics should arguably be the first step in vision data processing and analytics. Smoke, spatter, or large melt-pool geometry deviations can result from sub-optimal process parameters such as (but not limited to) deposition trajectory, velocity, feed-stock delivery rate, and energy source power [23, 22], while pixel saturation, obstructed field of view, or an out-of-focus lens are camera-related issues; such issues are often not considered, with most studies focusing on unrealistic laboratory-like conditions for observations. Sub-optimal process parameters and/or camera setup can lead to the occlusion of melt-pool features and the inability to use these images for further feature extraction, process control, or defect prediction. An image-based classifier for DED AM processes that is able to detect whether an image can be used for further information retrieval on melt-pool geometry based upon the visibility and presence of the melt-pool in the image is currently missing in literature.

The predictive performance of supervised ML models depends on the quality and size of the annotated training data-set. While data generation has become easier than ever with innovations in monitoring technologies, annotating unlabeled data can be labour intensive, difficult and time-consuming [149]. This is especially true in AM, where imaging-based process monitoring can generate high-dimensional data over long periods of time, which often requires manual annotation on the pixel level.

Active learning (AL) is a sub-field of ML focused on maximizing the performance of ML models by using the least amount of annotated training data. Instead of annotating

and training a model on a subset of the available data selected through uniform random sampling (RND), AL trains a ML model in an iterative way. At each iteration, the ML model is re-trained on the existing and newly labeled data instances, which have been selected by a query strategy and manually annotated by humans. This process is repeated until termination, or when the desired number of labeled instances is achieved [150].

The goal of the query strategy is to select unlabeled instances which will lead to the highest performance gain of the ML model. Uncertainty sampling (US) is a popular strategy which selects instances based upon the predicted class uncertainty [151], but can lead to the selection of redundant instances or outliers which do not add to the model performance. Although many efforts have been made to improve US [152], they often introduce additional computational complexity [153]. Furthermore, comparison shows that US still ranks near to top among existing query strategies [154, 155].

In this research work, the first goal is to present a novel query strategy named *Adaptive Weighted Uncertainty Sampling* (AWUS). This method combines US with random exploration of the instance space to prevent the selection of redundant instances. At each active learning iteration, a probability-mass-function (pmf) is defined, and sampled without replacement, to form a batch of unlabeled instances. The shape of the pmf is dependent on the change of the model predictions between AL iterations. Whenever the model does not change, it is expected to have explored the instance space enough such that it can focus on exploiting model knowledge. A large change on the other hand, is thought to be exemplar of large parameter uncertainty in the model, and exploration should be the focus. To achieve this, AWUS converges towards equal sampling probability, equivalent to RND, for large model changes, while near equivalent models between AL iteration assign a very large probability to the most uncertain unlabeled instances. The latter case converges towards US.

Additionally, in this research work, the second goal is to propose a new feature extraction and classification method via ML for in-situ quality prediction of DED processes. The classifier predicts, based on melt-pool visibility, whether an image, acquired through in-situ measurements, can be used for further quality assurance data evaluation, with such evaluation as out of scope for this work. The in-situ vision data sets of the AM process are often used for training purposes of ML models; such datasets typically include redundant images, as the process is repetitive in nature. In this work, we propose the use of AL via the AWUS method to significantly reduce the required annotation workload for training ML models.

The proposed active learning methodology AWUS is general in nature and can be applied to any ML task using a model capable of providing instance uncertainty. The

DED feature extraction and classification methods can be extrapolated to other AM processes where vision data is deployed to observe the interaction between an energy source and material. Overall, AWUS was tested to prove superior performance against state-of-the-art query strategies on 28 data-sets from the OpenML database and 8 DED in-situ imaging data-sets, each captured from a different DED machine, using the proposed feature extraction and quality classification method. The research contributions of this work are outlined in the manuscript as follows: Section 3.3 reviews related work in AL and melt-pool based ML in DED. AWUS and the DED feature extraction and classification methods are presented in Section 3.4. Experimental results are discussed in Section 3.5, followed by conclusions in Section 3.6.

3.3 Related work

3.3.1 Active learning

Supervised ML models require annotated data to train. A commonly-taken approach to train these models is to annotate a subset of the available data to reduce annotation time. Uniform random sampling, or RND, is a common data instance selection method, or query strategy, and is considered a passive learning approach. Active learning extends the annotation process by performing passive learning in an iterative way. At each iteration, a small number of unlabeled instances, called a batch, are selected and added to the already existing pool of labeled instances. Thereafter, the model is retrained on the new set of labeled data instances. This iterative approach allows the query strategy to exploit knowledge of the previously trained model for the selection of every new batch. Active learning has proven to regularly outperform passive learning, as less annotated data is needed to reach equivalent model performance [149, 155]. A visualization of the relation between passive and AL is provided in Figure 3.1.

Many AL algorithms, with different query strategies, have been proposed in the past. Recently, data-driven methods, which learn a query strategy from existing annotated datasets have been proposed [156, 157, 158]. A major drawback is however that annotated, and possibly large, datasets representative of the problem need to exist for training purposes on beforehand. We therefore focus on query strategy methods which do not require training. These methods can roughly be divided by the heuristics used to select instances for annotation. Two of the main categories include selection based upon instance informativeness and representativeness. Heuristics based on informativeness try to select instances which potentially lead to the largest decrease in the uncertainty of a

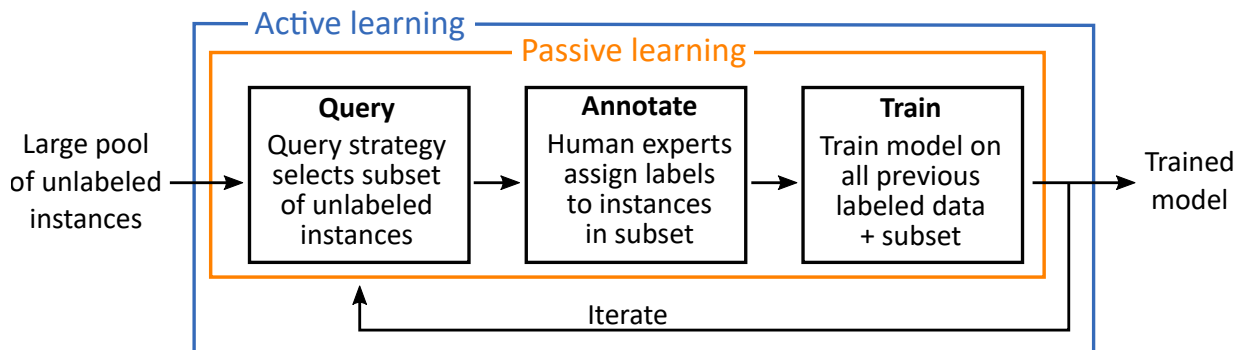


Figure 3.1: Active vs Passive learning. Passive learning is a feed-forward method where a supervised machine learning model is trained once on an annotated subset of the available data. Active learning uses feedback to select batches of instances in an iterative way. The query strategy in the active learning setting is able to exploit model knowledge of the previous iteration to select the next batch possibly leading to improved model training performance using less annotated data.

statistical model, while representativeness-based criteria seek to select instances which represent the underlying data distribution.

A further distinction for informativeness-based methods can be made by their frequency of model re-training. Using US [151, 159] for probabilistic classifiers, or maximum hyperplane margin for support-vector-machine classifiers [160, 161] requires to re-train the model once every AL iteration, while the query-by-committee methodology [162] requires multiple models to be re-trained. Other methods require model re-training proportional to the amount of unlabeled instances to evaluate the expected informativeness of each instance once labeled [163, 164, 165].

Methods seeking representativeness regularly use pairwise instance similarity as the basis for selection and are often used in combination with informativeness to select instances with high model uncertainty that are representative of the underlying distribution. Clustering and density weighted methods [166, 167] assign more weight to instances in dense regions, while diversity methods [168, 169] provide exploration by preferring the selection of dissimilar instances; others combine uncertainty, density and diversity [170, 171, 172].

The computational complexity and performance of multiple instance selection methods is reviewed in [173, 154, 152] and their results show that model re-training and instance similarity based heuristics lead to a drastic increase in computational time for

the selection of instances. These methods could therefore form a problem in ML, where instance similarity is difficult to define or when model training is time consuming; such is the case of image-based deep learning [174] and ML applications in AM, as further discussed in Section 3.3.2.

The introduction of randomness into the selection of instances can provide a computationally low cost alternative for instance similarity-based exploration. Furthermore, ill-defined instance similarity metrics would be avoided. Weighted uncertainty sampling was introduced by [175]; the method assigns a selection probability to each unlabeled instance, proportional to their classification uncertainty. Their method does however not have the ability to adjust the sampling probability weight of instances to focus more on exploration or exploitation. Furthermore, it does not adapt throughout the AL process.

The works by [176] and [177] used the class predictions between AL iterations to quantify ML model change. A low model change, is indicative of the exploitation phase of the active learning process, while a high model change, is related to the exploration phase. The authors used this metric as a probability to decide whether to use RND or US respectively. However, a major drawback is that either unlabeled data instances have the same probability of being selected whenever RND is chosen, or no probability of being selected, with the exception of the most uncertain one, for US.

The proposed method in this present work, *Adaptive Weighted Uncertainty Sampling*, called AWUS, combines weighted uncertainty sampling with the adaptability of model change and solves the problems of [175, 176, 177] highlighted above by combining approaches. This methodology will be explored on a variety of datasets from the OpenML database [178] and vision data from metal AM processes, specifically DED.

3.3.2 Vision-based process quality prediction in AM (DED)

Several methods for process quality prediction in metal AM processes based on melt-pool features have been proposed in the past [66]. Measuring the melt-pool geometry is useful for monitoring the process stability deviations, as well as for real-time control of AM processes. The extraction of the melt-pool boundary based on image processing heuristics such as pixel intensity thresholding, filtering, morphology operations, and edge detection [179, 180, 181, 182, 146] are popular due to their simplicity, but lack flexibility as they are tuned to work for a specific DED process setup, with specific camera settings, and expected melt-pool appearance. A more flexible and general approach is taken by [148], which uses a convolutional neural network (CNN) to perform melt-pool segmentation. However, the

downside is the need to annotate large amounts of images to train the ML model, which is time consuming since the melt-pool boundary needs to be manually determined by human annotators, on the pixel level.

The appearance of the melt-pool is also used to predict the melt-pool depth of the molten material as an indication of the melting mode physics phenomena. Over- or under-melting of the metal can lead to creation of porosity in the material due to the formation of cracks, gas-pores, or intra- or inter-layer overlap defects [55]. Examples of such efforts include the work by [183] which applies principle component analysis to extract and reduce appearance information from melt-pool images to a lower dimensional feature vector and a hidden Markov model to predict the penetration quality. [126], [129] and [125], on the other hand, use a CNN to extract features and classify melt-pool penetration from images directly. These works do however rely on quality datasets without interference from smoke, spatter or other phenomena obstructing the visibility of the melt-pool.

Perhaps the most interesting approach is taken by [128], as an image classification step is introduced before the prediction of the penetration level. They observed that saturated pixels in the images would lead to meaningless predictions. Therefore, they trained a CNN to distinguish between good and bad images before being fed into the penetration prediction network. A more sophisticated process and image quality classification method seems however to be missing in literature.

The proposed feature extraction and image quality classification method in this present work extends upon the concept captured in the work by [128], by classifying image quality into multiple categories based on the visibility of the melt-pool for DED processes. This method contributes to the field of data science, with an application in DED, by introducing a more general classification method based upon melt-pool occluding features as smoke, spatter and the plasma arc.

3.4 Methods

3.4.1 Active learning methodology and performance evaluation

This work considers the pool-based batch-mode AL setting (PBAL), which is described in Table B.1 of the Supplementary Material B, where a large pool of unlabeled data instances is available beforehand. This approach translates well to the AM field, where unlabeled instances in the form of in-situ acquired sensor data are often recorded during experiments, leading to pools of unlabeled data instances such as frames from video-recordings. The

amount of classes per dataset is considered variable, therefore binary- and multi-class classification problems are considered. At each AL iteration, a single or a number of unlabeled instances, called batch, are selected by a query strategy, labeled by human annotators and added to the existing pool of labeled instances. The updated labeled pool is then used to (re)train the ML classification model. The goal of the query strategy is to select the unlabeled instances leading to the highest gain in classification performance.

Initially, we assume the existence of a large pool of u unlabeled data instances $U = \{x_i\}_{i=1}^u$, a small labeled pool $L = \{(y_i, x_i)\}_{i=1}^l$ of l instances, the set C of unique class labels in $U \cup L$, a classification model m not trained on L , a set of human annotators H and a query strategy q . Additionally, a batch size b and termination condition t are provided controlling the AL execution. Although this work considers the batch size to be constant throughout the AL process, it is possible to select a different number of unlabeled instances at each AL iteration. The termination condition can be as simple as a limited number of AL iterations or based upon other criteria available during the process. The complete history of the pools and models of every AL iteration is available, this present work however considers the subset $U, L, m, \bar{m}$ to be the available prior information accessible by query strategy q to select a batch U_B at each AL iteration with m and $\bar{m}$ the previous and current model respectively. The annotators H are thereafter deployed to annotate U_B leading to labeled batch L_B . The updated labeled pool is used to retrain the classification model resulting in updated current model $\bar{m}$. This process is continued until termination or when no unlabeled instances are left. The last trained model $\bar{m}$ and labeled pool L are considered the results of the AL process and returned to the user.

In this present work, query strategy performance is evaluated through AL simulations on pre-annotated datasets using the procedure described in Table B.3 of the Supplementary material B. Each dataset is split into a training and validation set where AL is performed on the training set and ML model classification performance is evaluated on the validation set. The benchmark ML model performance, called BENCH, is the score achieved by the classifier on the validation set whenever trained on the complete pre-annotated training dataset. A trained model and sequence of labeled data instances L are the results from each active learning experiment. Using sequence L , active learning curves are created through retraining the classification model for every number of annotated data instances in L , as shown in Figure 3.2(a), and evaluating the classification performance on the validation set against BENCH. The distribution of the proportion of data instances needed to reach BENCH could take any shape. Using the mean and standard deviation of the distribution is therefore not used to compare active learning results, since it could provide misleading results in case different distributions are compared. The cumulative distribution provides insightful information on the distribution of results, as is displayed in Figure

3.2(b). A direct comparison between query strategies is possible for different proportions of simulations by reading the figure as; "A % of the simulations needed B (or less) % of the training data to reach BENCH for query strategy 1 while query strategy 2 needed C (or less) %". A single quantitative measure to compare query strategies is obtained by calculating the area under the cumulative curve (AUCC) in percent where a higher score relates to better AL performance.

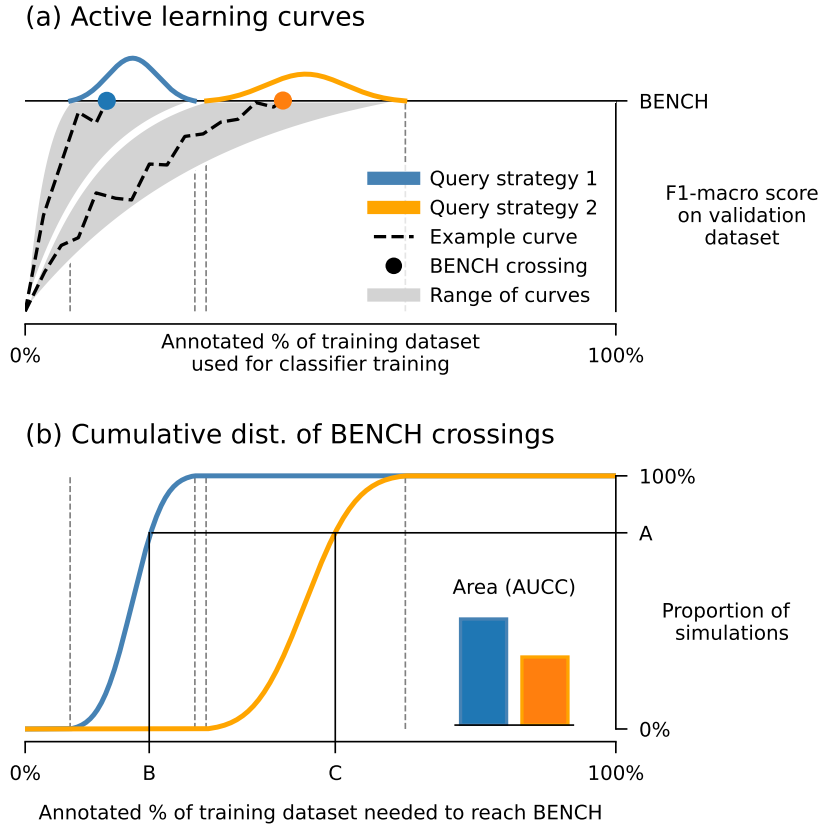


Figure 3.2: General query strategy performance evaluation strategy. (a) Active learning curves example where BENCH is the score achieved by the classifier when training on the complete training dataset. (b) Cumulative distribution of BENCH crossings. Read as: "A % of the simulations needed B(or less) % of the training dataset to reach an equivalent F1-macro score as if trained on the complete training dataset for query strategy 1 while query strategy 2 needed C (or less) %". The area under the cumulative curve (AUCC) is used to compare query strategy performance where a larger area is better.

3.4.2 Adaptive weighted uncertainty sampling

In this section the authors present a novel query strategy, *Adaptive-Weighted-Uncertainty-Sampling*, or AWUS, which is described in Table 3.1. The proposed method is adaptive, since it balances exploration and exploitation based upon the change of model predictions between AL iterations. This change, in combination with the classification uncertainty of the unlabeled instances, assigns a weight to each unlabeled instance. At last, these weights are turned into a pmf [184] which is sampled resulting in an unlabeled batch of instances to be annotated.

To better understand the algorithm, the definition of model change is important. The conditional probability of a label y , given data instance x , on the basis of a classifier architecture m trained on L is defined as $P(y|x)$. A decision function d which predicts the class y for a given instance x can then be described as follows:

$$d(x) = \operatorname{argmax}_{y \in C} P(y|x) \quad (3.1)$$

The previous and current decision functions d and $\bar{d}$ are available at each AL iteration since the previous and current classification models m and $\bar{m}$ are available. The difference between the predictions, which is related to model change, can be quantified using any metric able to define similarity. In this work we explore two metrics. The first metric is the cosine similarity metric s_c , which is used by the authors of [176] and [177], and defined as:

$$s_c = \frac{\langle D, \bar{D} \rangle}{\|D\| \|\bar{D}\|} \quad (3.2)$$

with $D = (d(x)|x \in X)$ and $\bar{D} = (\bar{d}(x)|x \in X)$ representing ordered sets of the real-valued previous and current class label predictions for all labeled and unlabeled instances $X = L \cup U$. The ratio similarity metric s_r is the second metric used in this work which is defined as:

$$s_r = \frac{N}{|X|} \quad (3.3)$$

with $N = \sum_{x \in X} \mathbf{1}_{[d(x)=\bar{d}(x)]}$ the number of equivalent predictions and $\mathbf{1}$ as the indicator function. Both s_c and s_r have positive range $0 \leq s \leq 1$ since $d(x) \in \mathbb{Z}^+$. Instead of weighting the sampling probability of each instance proportional to the classification uncertainty, as in [175], our method calculates an exponential weight e at each AL iteration,

which is defined by similarity s , to shape the pmf of the instances according to model change. We define exponent e as:

$$e = \frac{1}{1 - \max(s, \epsilon)} - 1 \quad (3.4)$$

with $\epsilon \approx 0$ as a very small number to prevent zero division. For our experiments we used $\epsilon = 1e^{-4}$. The exponent inversely scales s and is used to weight the classification uncertainty of each unlabeled instance. Although multiple metrics exist to quantify classification uncertainty [153], in this present work the authors use least confidence [151] for its simplicity, where instance uncertainty is defined as:

$$u(x) = 1 - \max_{y \in C} \bar{P}(y|x) \quad (3.5)$$

where $\bar{P}$ represents the conditional probability of model $\bar{m}$. Since the range $0 \leq u(x) \leq \frac{|C|-1}{|C|}$ is dependent on the number of possible classes in the dataset we introduce normalized uncertainty $n(x)$ according to:

$$n(x) = \frac{|C|}{|C| - 1} u(x) \quad (3.6)$$

with range $0 \leq n(x) \leq 1$. The exponent e and normalized uncertainty $n(x)$ are thereafter used to assign a weight $w(x)$ to each unlabeled instance x by

$$w(x) = (n(x) + 1)^e \quad (3.7)$$

with range $1 \leq w(x) \leq 2^{\frac{1}{\epsilon}-2} \approx \infty$. At last, instance weights are transformed into pmf $p(x)$ by:

$$p(x) = \frac{w(x)}{W} \quad \forall \quad x \in U_s \quad (3.8)$$

with $U_s \subseteq U$ the subset of available unlabeled instances during batch construction. Normalizing constant $W = \sum_{x \in U_s} w(x)$ scales the weights w such that $\sum_{x \in U_s} p(x) = 1$ and $p(x)$ resembles a pmf. The relation between instance uncertainty $u(x)$ and the probability of being sampled from $p(x)$ as a function of s is showcased in Figure 3.3.

AWUS is applicable to any ML dataset or task, with the only constraint being a model capable of providing instance uncertainty. No definition of instance similarity for instance exploration is needed, AWUS is therefore well suited for AL tasks where instance similarity can be difficult to define, such as computer vision in AM, and DED particularly.

Table 3.1: Adaptive Weighted Uncertainty Sampling (AWUS). A batch U_B of unlabeled data instances is selected/sampled from the pool of unlabeled data U .

Input	U - Pool of unlabeled data instances L - Pool of labeled data instances m - previous model $\bar{m}$ - current model C - Set of unique class labels b - Batch size
Output	U_B - Batch sequence of unlabeled instances
1	$e \leftarrow m, \bar{m}, U, L, C$ // Construct exponent using eqs.3.1,3.2,3.3,3.4
2	$U_B \leftarrow \emptyset$ // Empty batch
3	while $ U_B < b$ do
4	$U_s \leftarrow U \setminus U_B$ // Get available unlabeled instances
5	$p \leftarrow \bar{m}, C, e, U_s$ // Construct pmf using eqs.3.5,3.6,3.7,3.8
6	$x \xleftarrow{\text{sample}} p$ // Sample unlabeled instance
7	$U_B \leftarrow U_B \cup x$ // Add unlabeled instance

3.4.3 DED process quality classification based on melt-pool visibility

In this work, we consider 8 different wire-fed directed energy deposition processes with vision-based in-situ measurement setups using camera's. The location and orientation of the camera is assumed to be variable. From the acquired DED datasets, which are further discussed in Section 3.4.4, recurring features are observed to be present in the images. An example image with all of these features, or objects, is shown in Figure 3.4.

The prediction of process quality from imaging data is dependent on the quality of acquired sensor data. Since the melt-pool geometric features are used for the prediction of melting mode phenomena, defects, deposition geometry, melt depth, cooling and solidification rates, as discussed in Section 3.3, the ability to observe and measure the melt-pool geometric features is crucial. Based on melt-pool visibility, every image is classified as either (i) no melt-pool, (ii) occluded melt-pool or (iii) segmentable melt-pool based on the presence and visibility of features in the field-of-view of the camera. The definition of the three classes, along with reasons to assign an image to a specific class, are provided in Table 3.2.

In the proposed methodology, each image is compressed to a lower dimensional feature

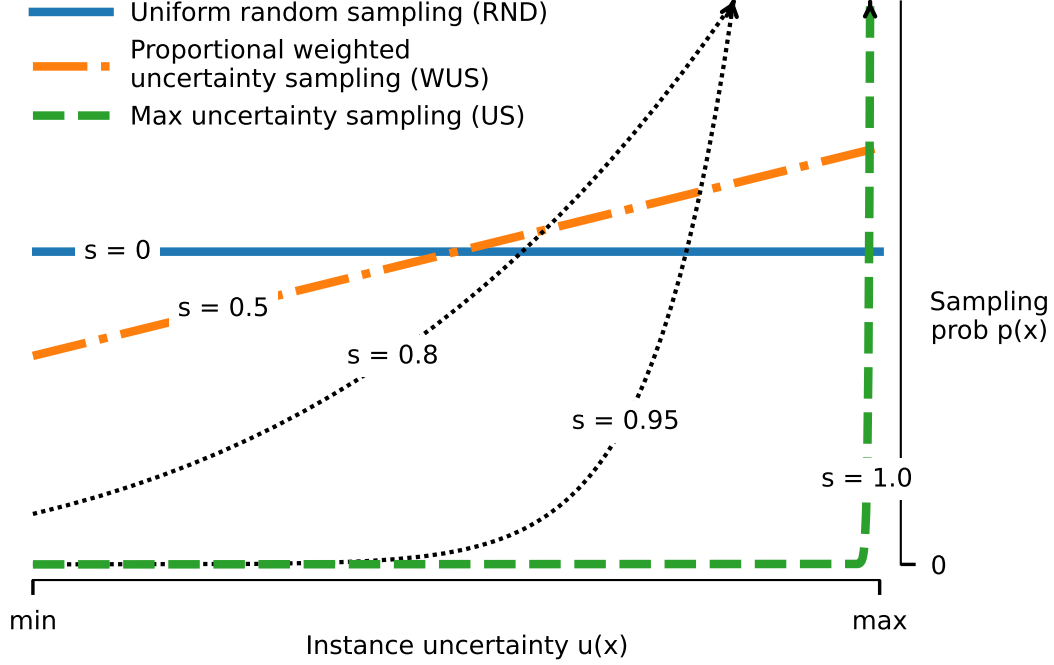


Figure 3.3: Showcasing of the relation between similarity s and instance sampling probability $p(x)$ for various values of s . The particular values $s = 0, 0.5, 1$ result in exponents $e = 0, 1, \infty$ leading to uniform random sampling (RND), proportionally weighted uncertainty sampling (WUS) and max uncertainty sampling (US) respectively.

vector to reduce computational complexity, extract features related to the visual signature of the objects observed in Figure 3.4, reduce the sensitivity to different lighting conditions, and ensure invariance of response to rotation and position in the field-of-view. To reduce sensitivity to different lighting conditions, a min-max scaling is used on every image I according to:

$$N = \frac{I - \min I}{\max I - \min I} \quad (3.9)$$

resulting in scaled image N such that $\max N = 1$ and $\min N = 0$. Thereafter, we construct feature vectors for each image by concatenating the histogram of pixel intensities and histogram of pixel gradient magnitudes. Histograms provide information on the distribution of pixel values, therefore being invariant to rotation or position. Furthermore, The value of each histogram bin (or feature) is achieved by calculating the

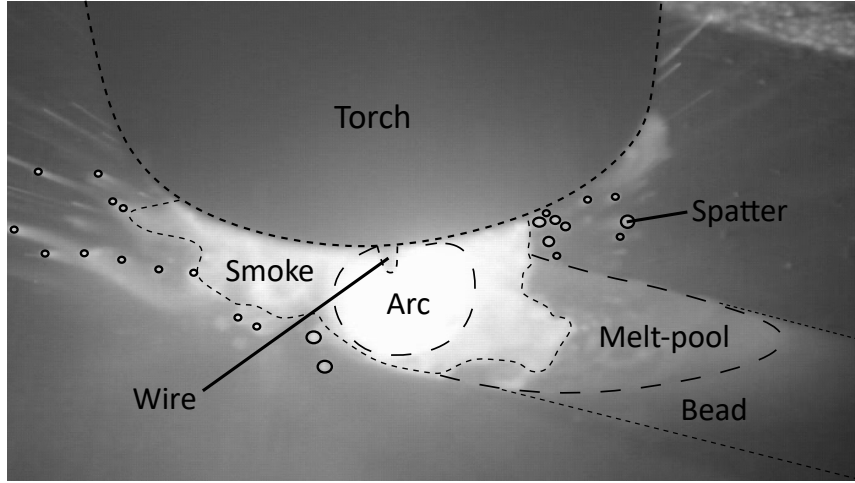


Figure 3.4: General recurring objects observed in images from DED datasets

Table 3.2: DED image classification

Classification	Reasons to classify
No melt-pool	Melt-pool not visible (process did not start yet, has ended, is interrupted or obstructed) or outside camera field-of-view.
Occluded melt-pool	Melt-pool visible but boundary obstructed through spatter, smoke, arc, torch, bead, wire, pixel saturation, bad lens focus.
Segmentable melt-pool	Melt-pool visible and boundary not obstructed.

number of pixel values in a value range. Each bin is assigned a different value range. A smaller range of pixel values assigned to each bin requires more bins to capture the complete range of pixel values. This provides the ability to increase or reduce the histogram size, thereby controlling the number of features. In the following paragraphs, the rationale and mathematics behind the feature extraction methodology are discussed. Conformation of our approach as well classification results are further discussed in Section 3.5.1.

Each melt-pool class, as presented in Table 3.2, is expected to show, on average, a different image signature in terms of pixel intensities distribution. The distribution of pixel intensities in images N classified as "No melt-pool" is expected to be relatively uniform compared to the other classes due to the absence of higher intensity process signatures such as the plasma arc and spatter. The "Segmentable melt-pool" images show larger

differences in pixel intensities, since low intensity pixels belong to the background, while high intensity pixels belong to the melt-pool, arc, bead and other bright features. Occluding image signatures, such as smoke and spatter, are of equal or lower intensity compared to the plasma. Furthermore, these features tend to blend the images due to the smoothing effect of process, setup, and sensor phenomena. The distribution shape of pixel intensities for "Occluded melt-pool" images is therefore expected to show more differences between the number of low and high intensity pixels than the "No melt-pool" class but less than the "Segmentable melt-pool" images. To capture the intensity distribution of scaled images N , a histogram of intensities $hist(N, b_N)$, is computed for every image leading to a vector of b_N features (bins), with bin edges $(0, \frac{1}{b_N}, \dots, 1)$.

Besides intensity information, the magnitude of gradients' distribution in each image, capturing edges, is used to further distinguish between the classes. Images belonging to the "Segmentable melt-pool" class are generally sharp without the presence of many occluding features blending the images. Sharp edges at the melt-pool boundary are therefore preserved. This in contrast to the "Occluded melt-pool" images where the melt-pool boundary could be occluded by process, setup and/or sensor phenomena. The edge gradient intensities are therefore expected to be lower as well. The distribution of gradients in scaled images N belonging to the "No melt-pool" class will generally show a more uniform distribution of gradient intensities. The differences in distribution shape between the classes is therefore expected to be comparable to the distribution of pixel intensities. The magnitude of gradients for each scaled image N is calculated using Sobel operators as follows:

$$S_x = [1 \ 2 \ 1]^T * ([1 \ 0 \ -1] * N) \quad (3.10)$$

$$S_y = [1 \ 0 \ -1]^T * ([1 \ 2 \ 1] * N) \quad (3.11)$$

$$G = \sqrt{S_x^2 + S_y^2} / \sqrt{2} \quad (3.12)$$

with "*" the convolutional operator leading to image of gradient magnitudes G . The magnitude is divided by $\sqrt{2}$ to ensure $\max G \leq 1$ and $\min G \geq 0$. The resulting feature vector H is constructed by concatenation of the histogram of normalized intensities $hist(N, b_N)$ and histogram of magnitude of gradients $hist(G, b_G)$ according to:

$$H = (hist(G, b_G), hist(N, b_N)) \quad (3.13)$$

The choice for the number of bins b_N and b_G is further discussed in Section 3.5.1. Large differences in scale are possible with the use of histograms which could possibly lead to

difficulties during classifier training. We therefore apply a normalized natural logarithm transformation by:

$$x = \frac{\ln(H + 1)}{\ln(\sum H/2 + 1)} - 0.5 \quad (3.14)$$

such that $\max x \leq 0.5$ and $\min x \geq -0.5$. The resulting feature vector x is calculated for every image and used for classifier training as will be further discussed in Section 3.5.1. A visualization of the feature extraction process is displayed in Figure 3.5.

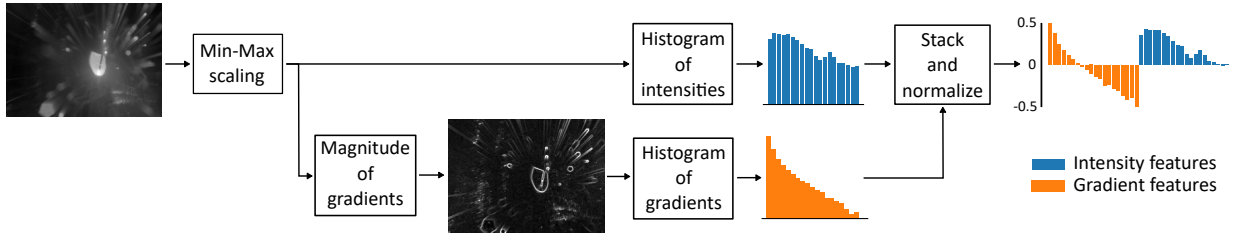


Figure 3.5: Feature extraction of in-situ images from DED processes. Min-max scaling (equation 3.9) of input image, magnitude of gradients (equation 3.10), histogram of pixel intensities and gradients (equations 3.13), stacking and normalizing (equation 3.14) lead to feature vector x , with b_G gradient features and b_N intensity features.

3.4.4 Experimental setting

Our proposed novel active learning query strategy, as discussed in Section 3.4.2, is compared against 5 state-of-the-art query strategies and RND, which serves as a baseline performer, using the performance evaluation method described in 3.4.1. Related to the discussion in Section 3.3, the following query strategies are selected:

- Uncertainty sampling (US), which selects unlabeled instances with the highest uncertainty [151, 153].
- Proportionally weighted uncertainty sampling (WUS), which samples unlabeled instances weighted by their model prediction uncertainty [175].
- Uncertainty, density and diversity sampling (UDD), which selects uncertain instances in dense regions far away from other labeled instances (diverse) using the cosine instance similarity metric [172].

- Uncertainty sampling with a random exploration component (EGA), which randomly chooses to use uncertainty or random sampling at each query with a probability defined by model change [177].
- Another combination of uncertainty and random sampling (BEE), with a slightly different selection method compared to EGA [176].

A diverse set of 36 datasets is selected to evaluate and compare AL performance of the query strategies and AWUS. The majority, 28 datasets, are acquired from the OpenML database [178] with most of them originating from the UCI repository [185]. These datasets are commonly used in active learning research [155, 154] and selected based upon popularity. The other 8 datasets are created through feature extraction and annotation of 8 in-situ video-recordings, each acquired from a different DED process, as described in Section 3.4.3. An example image per DED dataset and class is visualized in Figure 3.6. The number of images assigned to each specific class, in each dataset, is displayed in the right lower corner of each example image. Specifics for each dataset are presented in Table B.2 of the Supplementary material B. The annotation protocol used to link each image with a specific class is provided in Figure A.1 of Supplementary material A.

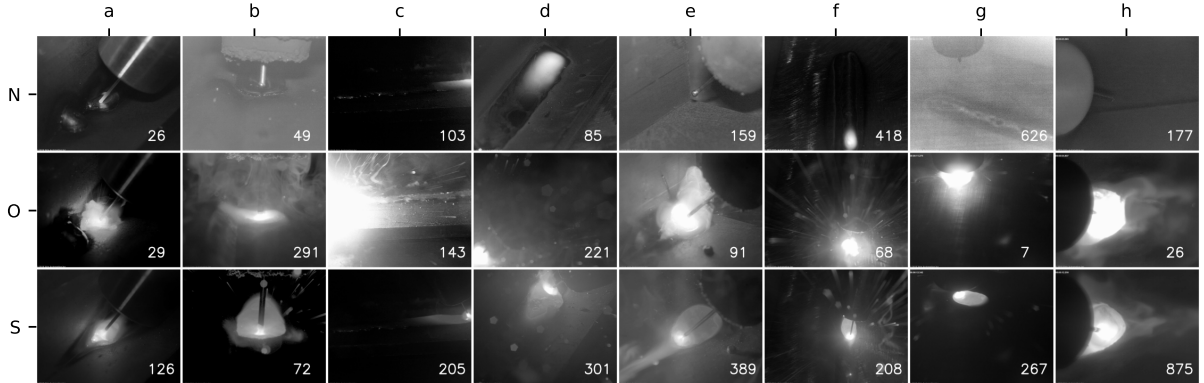


Figure 3.6: Image examples for every DED dataset [a,h] and class [N,O,S]. Each image in every dataset is manually labeled as either No(N)-, Occluded(O)- or Segmentable(S)-melt-pool. Number of images per class shown in corner of image. Additional information can be found in Table B.2 in the Supplementary material B.

All DED datasets are acquired through in-situ video recordings of steel wire-fed DED deposition systems on steel using a Xiris XVC-1000 camera at 55 frames per second with a

50mm Ricoh FL-CC5028-2M optical lens creating gray-scale images of 1024x1280 pixels. A Midwest optical systems BP635-27 red light bandpass filter is used during data acquisition of datasets [b,e,f] (see Figure 3.6) while the rest is acquired without any filter. Datasets [a,d,e] are created using T-joint deposition while the rest uses a flat surface. The camera view is leading the deposition in [a,b,e,f,h], trailing in [d] and from the side in [c,g].

Logistic Regression (LR) [151, 165], support vector machines (SVM) [159, 160, 161, 154, 164, 186], Gaussian naive bayes (GNB) [163, 169] and Random Forest (RF) [172, 187] are popular classifiers within active learning research. In this present work, all 4 classifiers were used for the comparison with the active learning and DED classification performance experiments. For the RF, 10 decision trees were used and a linear kernel for the SVM. The F1-macro metric was used to evaluate classification performance. This metric can be interpreted as a weighted average of the precision and recall for each class, and is appropriate for multi-class classification problems. The definition of conditional probability $P(y|x)$ of a class label y given data instance feature vector x is crucial for the workings of many active learning query strategies, as discussed in Section 3.4.2. For details regarding the implementation of the classifiers, definition of conditional probability and the F1-macro metric please refer to the Scikit-learn [188] ML library version 0.24. All experiments are implemented in Python 3.9 and computations are performed on a Intel i7-9700K CPU.

3.5 Results and Discussion

This section provides experimental simulation results for (i) the proposed DED classification methodology as described in Section 3.4.3 and (ii) the proposed active learning query strategy AWUS as described in Section 3.4.2.

3.5.1 DED feature extraction and classification results

To determine the influence of feature vector construction, as discussed in Section 3.4.3, on classification performance, we performed 1000 repeated experiments for each classifier, DED dataset and intensity and gradient features combination. For each experiment, each annotated dataset is divided into a 50/50 % training and validation set split using uniform random sampling. All images in both the training and validation set are thereafter turned into feature vectors of sizes 4, 8, 16, 32, 64, 128 and 256 for 3 feature balance cases; (i) 100% gradient features, such that the all features in the feature vector originate from the magnitudes of gradients histogram, (ii) 100% intensity features, such that only features

from the histogram of intensities are used, or (iii) 50/50% gradient and intensity features, where equal size histograms are used such that $b_G = b_N$. Each classifier is thereafter trained and evaluated for every feature vector size and case.

The F1-macro classification performance results over all classifiers and datasets is presented in Figure 3.7(a). Although feature vectors consisting out of 100% intensity histogram features perform better than 100% gradient histogram features, the combination of both intensity and gradient features provides superior performance for an equal amount of features. For feature vectors with more than 4 features a 50/50% contribution of gradient and intensity features consistently outperforms the others, with a median F1-macro score of approximately 90%.

For all DED datasets, the 50/50% gradient and intensity 16-bin feature vector were selected for further analysis. This amount of features is chosen as a trade-off between size and performance. Detailed performance figures for each classifier and DED dataset using this type of feature vector are provided in Table B.2 of the Supplementary material B. To test the proposed feature extraction approach as described in Section 3.4.3, the authors have shown the distribution of the values of the features in all 16-bin feature vectors for each class in Figure 3.7(b). The results confirm the hypothesis as the different signatures in the lower, middle and higher intensity and gradient regions are indeed showing the expected differences between the classes as discussed in Section 3.4.3.

The proposed feature extraction and classification method for DED processes shows to be an effective tool in classifying DED images based upon the visibility of the melt-pool. Our feature extraction method is performed on an image-by-image bases without the need to normalize based on a global dataset mean and standard deviation. As a result, our extraction method is therefore easy to implement for real-time applications as a feature vector can be generated whenever the image is acquired from the sensor. Although not investigated in this research, other ratios between the number of gradient and intensity features, or different feature extraction methods and classifiers can possibly lead to even better results; this can be an avenue for future work.

3.5.2 Active learning results on reference dataset

To clarify and visualize the difference between query strategies, the authors performed 10000 active learning simulations for each strategy using a batch size of 32 on a newly created reference dataset, named the "Horizon dataset", as presented in Figure 3.8. This binary dataset (two classes) has 1800 data instances with 2 features (x and y), where 225 points belong to the yellow and 1575 points to the blue class leading to a class imbalance

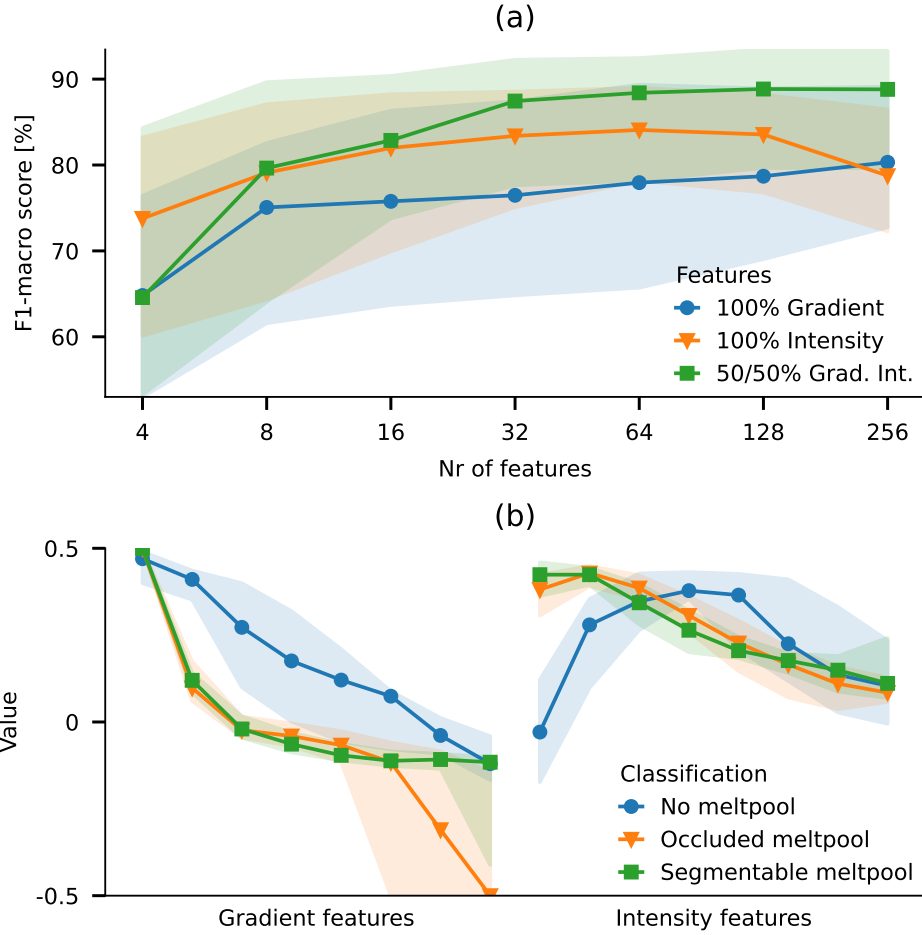


Figure 3.7: DED feature extraction and classification performance. All transparent areas correspond to the 25 and 75 data percentiles, while solid lines are the medians. (a) F1-macro score distribution over all classifiers combined on validation set against the number of features in each feature vector. (b) Distribution of 16-bin feature vector (50/50% grad. int) values for each class. The first 8 Features of each feature vector hold gradient features and the last 8 hold intensity features.

ratio of 1:7. Half of the data instances are positioned in a regular grid between -1 and 1 and a dense cluster of 900 instances in the yellow class provides the other half. A F1-macro classification score of 100% could be achieved by predicting the yellow class when $y > -0.5$ and blue otherwise. Class separation based on clustering is impossible for this dataset as the decision boundary lies within the regular grid. Exploration of the instance space and exploitation of model knowledge is therefore necessary to achieve good classification performance.

RND selects many instances in the majority class cluster. Since the actual decision boundary is far from this cluster, these labeled instances do not add to AL performance. US selects redundant instances close to the decision boundary of the current model. Initially however, the model has been trained only on a few labeled instances, leading to incorrect predictions and the selection of instances far from the actual decision boundary. WUS improves RND by incorporating instance uncertainty. The proportional weighting is however static throughout the AL process, therefore not exploiting model knowledge enough at later AL iterations. BEE and EGA randomly choose between random and uncertainty sampling based on a probability defined by model change between AL iterations. BEE uses less random sampling compared to EGA. Early AL iterations however show that either instances are selected in the majority class cluster when random sampling is chosen, or instances close to the decision boundary of the current model. Therefore combining both problems of random and uncertainty sampling. UDD selects instances near the cluster and across the instance space but lacks fast convergence towards the actual decision boundary. AWUS, the proposed algorithm by the authors, is evaluated using the cosine similarity metric (AWUS-C) and the ratio similarity metric (AWUS-R) as described by Equations 3.2 and 3.3 in Section 3.4.2. Both methods initially select unlabeled instances similar to RND and gradually adjust towards US at later AL iterations.

AWUS-C and AWUS-R outperform all other methods and reach a 99.88+ and 100% F1-macro classification score in 6 AL iterations by annotating only 11% of the unlabeled dataset, regardless of the initial conditions. Computationally, RND is most efficient, followed by WUS, AWUS-R, AWUS-C, US, EGA, BEE and UDD.

3.5.3 Active learning results on OpenML and DED datasets

Additionally, the performance of each query strategy is evaluated by performing active learning simulation experiments on the 36 real-world pre-annotated OpenML and DED datasets as described in Section 3.4.4. The exact simulation procedure is described in Table

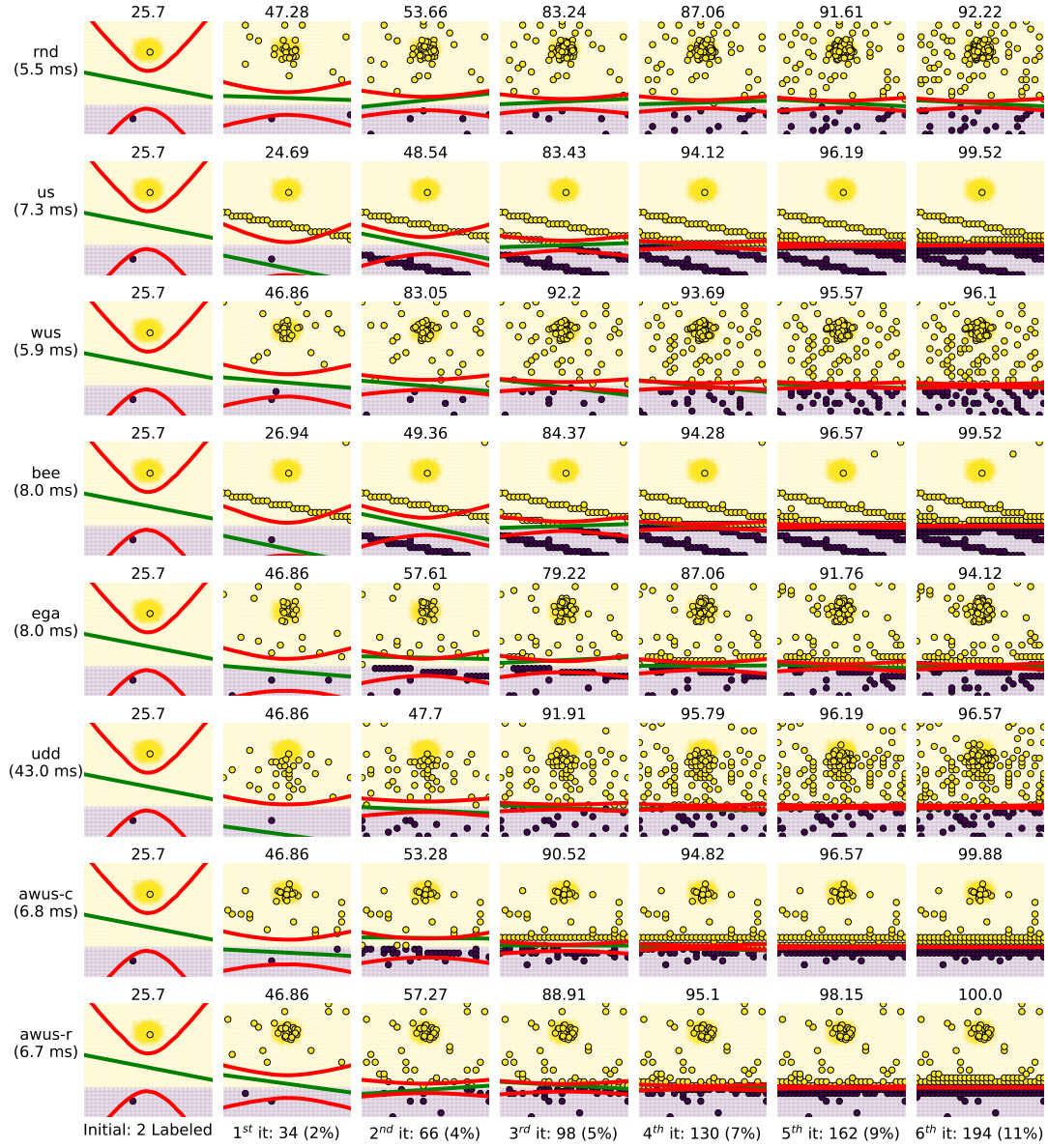


Figure 3.8: Results from 10,000 AL simulations with a linear SVM on "Horizon." Two initial labeled instances (column 1) is followed by 6 AL iterations (columns 2-7). Each iteration shows the lowest F1-macro score and average time. Red lines are the 95% decision boundary range; green lines show one simulation's boundary. White and black dots denote unlabeled and labeled instances.

B.3 of the Supplementary material B. We continue by providing a high-level overview of the procedure.

Each dataset is randomly split into a 50/50% training and validation set while maintaining the class balance ratio of the complete dataset in both sets. A single labeled instance per class from the training set is randomly selected leading to labeled pool L_0 . The unlabeled pool of instances U_0 is created by selecting all instances (without labels) from the training set which are not already present in the labeled pool. To investigate the effect of batch size on the performance of each query strategy, active learning is performed for batch sizes 1, 4, 16 and 64. For each combination of batch size, classifier and query strategy active learning is performed on the same initial datasets L_0 and U_0 . Since many combinations of splitting the dataset and initial known labeled instances exist, which might affect classification performance, we perform 1000 repeated active learning simulations, with different random selections of instances, on every dataset for all combinations of batch size, query strategy and classification model.

Overall, AL results over 4 batch sizes, 4 classifiers and 36 datasets, repeated 1000 times per query strategy is displayed in Figure 3.9. Each query strategy is sorted by their resulting AUCC score. AWUS-C (cosine similarity) and AWUS-R (ratio similarity) are the winners overall, as they outperform all other query strategies, not only in AUCC value, but also by examining the cumulative curve as less annotations are needed compared to the others at every proportion of simulations.

The performance per classifier and dataset is shown in Figure 3.10. The choice of classifier has a large effect on the distribution of active learning results. The AUCC evaluation method proposed in this work, with the use of cumulative curves, is therefore an appropriate choice as discussed earlier. AWUS-C and AWUS-R are not only outperforming most query strategies on the DED as well as OpenML datasets, but also the most robust, as other strategies show mixed performance and a high sensitivity to the usage of different datasets and classifiers.

The influence of batch size on the performance of query strategies per classifier is displayed in Figure 3.11. As expected, the overall AL performance decreases with an increase in batch size due to the less frequent model updates throughout the AL process. AWUS, and its variants however, are least affected by an increase in batch size compared to the other query strategies. AUCC scores for each classifier, dataset and query strategy are presented in Table B.4 in the Supplementary material B.

In real-world AL applications, larger batch sizes are normally preferred over smaller batch sizes [152, 189]. This way, the number of times the model needs retraining (which can be time-consuming) can be reduced. Especially in scenarios where multiple annotators

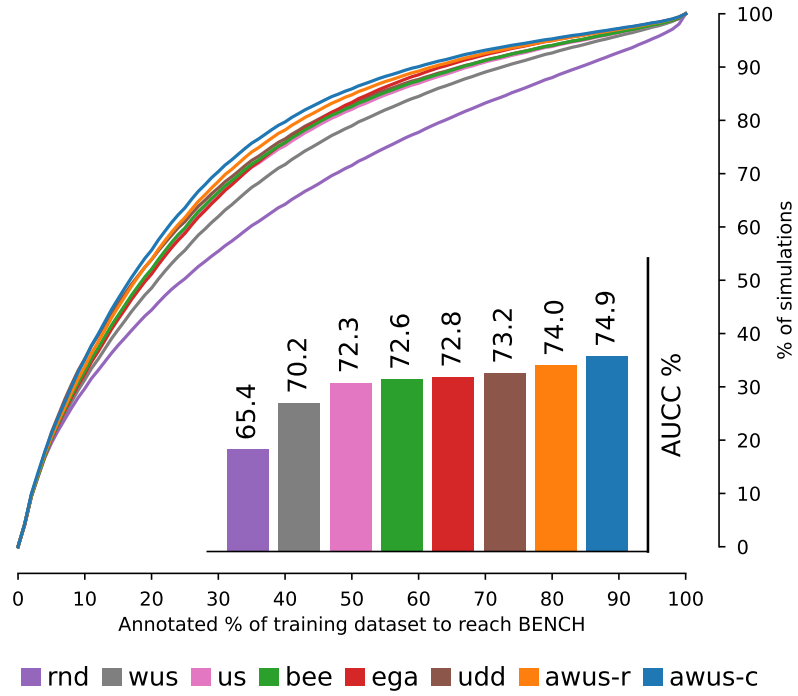


Figure 3.9: Active learning performance comparison of state-of-the-art query strategies and AWUS over all 36 datasets, 4 classifiers and 4 different batch sizes. Each parameter setting repeated 1000 times with different dataset splits (50/50% train-test) and initial conditions. See Figure 3.2 for evaluation metric details on AUC and BENCH.

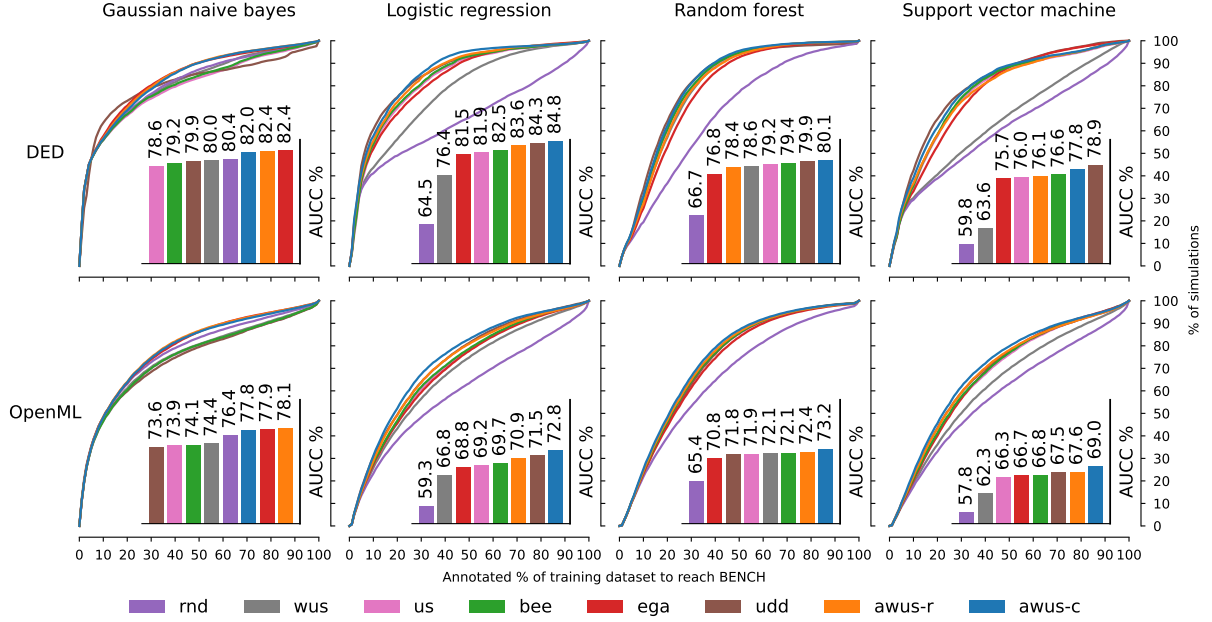


Figure 3.10: Active learning performance comparison of state-of-the-art query strategies separated by dataset source (DED or OpenML), and classifier. See Figure 3.2 for evaluation metric details on AUCC and BENCH.

are involved, since single instance selection would require each annotator to wait for the model to retrain after another annotator selected an instance to be labeled. Larger batch sizes allow multiple annotators to annotate in parallel on the selected batch of instances. Furthermore, the low computational cost of AWUS, since random instead of similarity based exploration is used, allows for faster AL iterations. Pairwise instance similarity can be difficult to define. Particularly in image processing where a definition of image similarity is often based upon the objects present in the image rather than pixel value distance [190]. As discussed in Section 3.4.3, DED image feature extraction and classification is performed based upon the visibility of the melt-pool. The performance of similarity based AL query strategies is dependent on the feature extraction method while AWUS, and other methods incorporating random exploration, can operate solely on model uncertainty. This can be useful in deep learning based ML methods where feature extraction and classification are embedded into a single neural network.

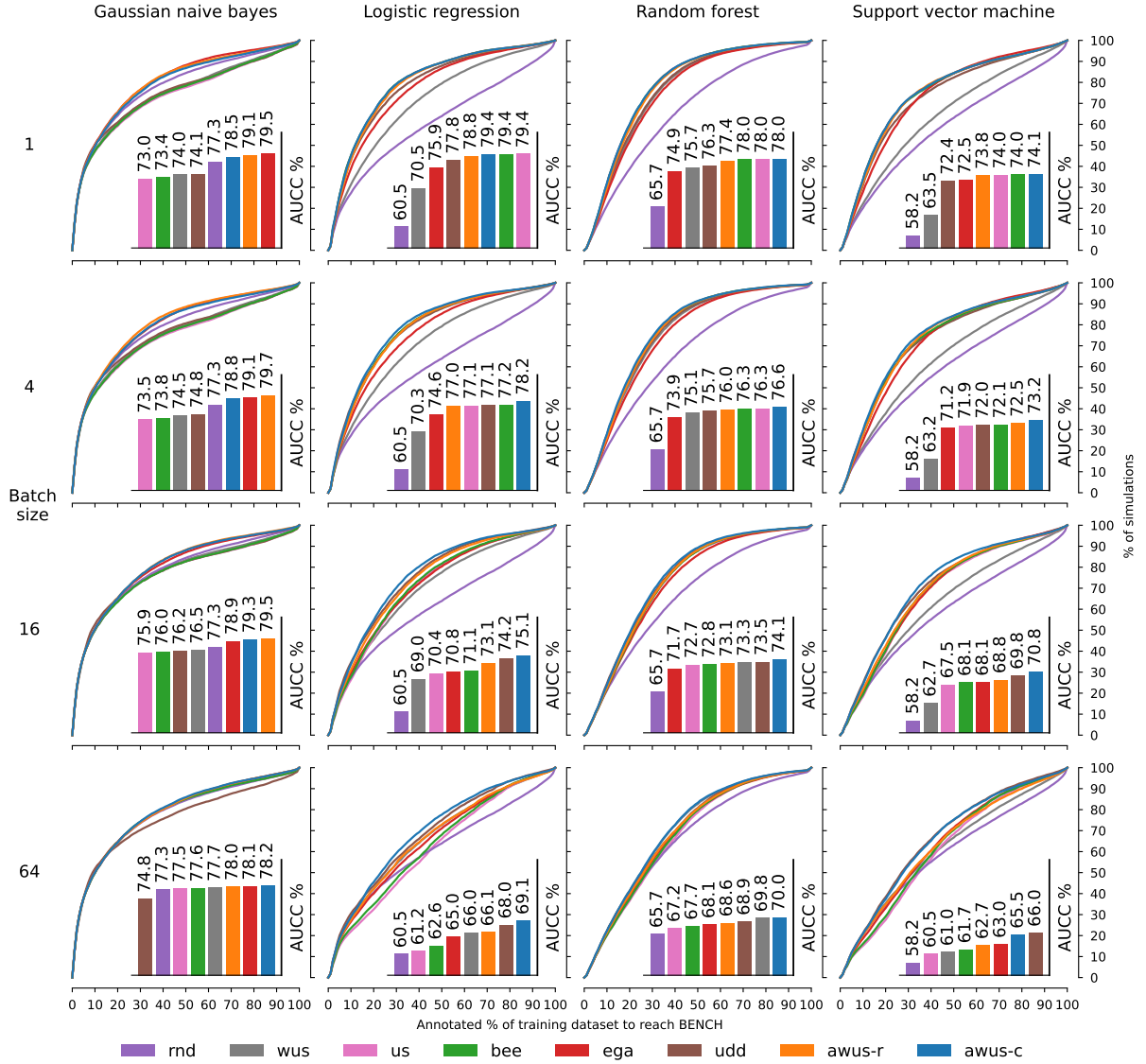


Figure 3.11: Active learning performance comparison of state-of-the-art query strategies separated by batch size (1,4,16,64) and classifier over all 36 datasets. See Figure 3.2 for evaluation metric details on AUCC and BENCH.

3.6 Conclusions

This work presents a novel active learning method AWUS which adaptively balances random exploration with uncertainty based exploration using model change. AWUS outperforms the state-of-the-art and is computationally low-cost. Since random exploration instead of instance similarity based exploration is used, feature representation does not affect active learning performance. Our method is therefore especially suited for image-based machine learning models where image similarity is difficult to define, such as in-situ process quality prediction for directed-energy-deposition (DED) additive manufacturing.

AWUS outperforms the state-of-the-art across 36 datasets from different domains (UCI, OpenML) on 4 batch sizes (1, 4, 16, 64) over 4 different classifier architectures (Logistic Regression, Support Vector Machine, Random Forest, Gaussian Naive Bayes) as the required amount of annotations needed to for training supervised machine learning models is reduced by 20 to 70% in 90% of the simulations.

Furthermore, we present a novel process quality classification method for DED processes based on the visibility of the melt-pool boundary in in-situ acquired imaging data. Our DED feature extraction and classification method is evaluated on 8 datasets acquired from different DED machines showing a median F1-macro score performance of 90+%.

Chapter 4

The impact of domain shift on deep learning model performance in directed energy deposition process quality prediction¹

4.1 Preface

Additive manufacturing (AM) processes can generate a wealth of in-process data. Current machine learning (ML) modelling efforts focused on informing quality of directed energy deposition (DED) processes are often created using small non-diversified datasets with significant similarity between training and validation data. This can potentially provide an unrealistic view on model performance in real-world operating scenarios, where input data can differ significantly from the training data. Although domain adaptation methods adapt existing models to new environments, they are inoperative in real-time quality prediction on edge devices, where re-training is impossible, and there is insufficient available target-environment data. A detailed definition of potentially experienced environment changes and their effect on machine learning performance is however missing in the AM and DED literature but could provide insights into the operability of ML models with and without the need for domain adaptation. In this work, we introduce a novel diverse DED dataset that enables training and validation across

¹Chapter is adapted from an article submitted for publication

multiple newly defined environmental changes typically experienced in DED. We evaluate a multitude of modern ML architectures and provide insights into the applicability of the models considering cross-environment performance and inference latency.

4.2 Introduction

Metal additive manufacturing (AM) is a promising smart manufacturing technology class due to its potential to reduce the product supply chain, from design to physical component, with increased customization possibilities and material utilization, thus leading to a smaller operational footprint, and on-demand manufacturing [191]. Directed energy deposition (DED) is a type of metal AM process where an energy source, typically a laser, melts a feed-stock material, typically powder, that solidifies to a substrate to build parts or features. DED processes can generate a wealth of in-process data across spatial, temporal, spectral, and radiometric scales of resolution. The input parameter space such as design, process parameters, material characteristics, and the associated post-process part characterization measurands are data streams which can be leveraged for extracting intelligence pertaining to part quality and process performance. Advancements in sensor technologies and data acquisition methods have led to a drastic increase in AM data available before, during, and after part fabrication, leading to new opportunities for knowledge discovery and quality improvement [192]. Recent efforts to improve process repeatability and part quality are focused on learning as much as possible about parts and processes through in-process monitoring and analytics [193]. Machine learning (ML) methods especially, have gained the attention of the AM and DED community, for the prediction of part quality and defect detection throughout the AM development cycle [194].

A recent thorough review on machine learning defect detection in metal AM has been conducted by Fu et al. [195] where studies on geometrical defects, porosity, cracks and incomplete fusion detection are summarized for several ML architectures and sensor data types. Although the overall average classification accuracy of 90+% across the 56 studies shows promise, the applicability and operability of these models is questionable as: (1) each study uses a single hardware setup consisting of a specific AM machine and data acquisition system; (2) a single or small finite set of experiments are conducted; (3) a single or limited set of different process parameters or machine settings are used; (4) All studies use uniform random sampling to divide their dataset into a training and validation set; (5) a robust justification of model selection is typically missing. Whether these models perform well in practical applications, with changes to the operating environment relative to the training data, such as but not limited to, different process parameters, a different

AM machine, a different part to build, changes in air-density, is unclear. A ML model trained and validated on data coming from the same data distribution, where a dataset is split into a training and validation set using uniform random sampling, only provides insights into model performance on that particular dataset and experimental setup.

The vast number of process parameters [41], material options [196], AM technologies [191], sensor technologies [197], and furthermore the possibilities and differences in input-data distribution are extensive. ML model training and validation is currently impossible for all of these possibilities due to time and cost constraints [198], leaving only room for a limited amount of AM experiments and therefore, in-situ data acquisition campaigns.

Transfer learning (TL) is a branch of ML focused on re-using a pre-trained model for a new task, with the potential to reduce the amount of data needed to re-train a model for the target environment, and has gained the interest of the AM community for the aforementioned reasons. The state of TL in AM has recently been reviewed by Tang, Dehaghani and Wang [199], and they concluded that the amount of training data in the target environment could be reduced to 20-70% of the data in the source environment. Although TL could reduce the amount of experiments needed to develop new ML models, it does not provide the tools or insight into ML model performance under changes in the operating environment it has not observed during training. The studies in [199] and [195] are therefore burdened by the same problems as only uniform random sampling is used to split training and validation data. Access to target data is required for TL, but often not practical in AM, and specifically DED, in case of on-demand manufacturing. TL would require experiments, possibly data annotation, and model re-training, before it can be used in the target domain, possibly for every novel manufactured part, and every machine.

The application, and predictive performance, of trained ML models on unseen data from different AM setups, process parameters, or builds is presently under-explored. If existing ML models could be used beyond their own specific setup, redundant model training, and TL, could be prevented. The creation of new supervised ML models requires the acquisition and annotation of new training and validation data, as well as model re-training which could be very inefficient, expensive and time consuming [145]. Training and validating ML models on datasets coming from a different input-data distribution could provide insight into ML model performance under different operating conditions. The effect of such a shift in input-data distributions, or domain shifts, on ML model performance has however yet to be investigated within AM and is part of the goal of the present study.

To motivate the exploration of novel ML architectures to improve performance under domain shifts, the evaluation of the existing state-of-the-art (SOTA) supervised ML model architectures under different domain shifts is imperative; such is the focus of this present

work. Studying the effect of multiple domain shifts on ML model performance requires a rich and diverse enough dataset such that a sufficient number of different training and validation splits can be generated to simulate and validate the different domain shifts.

This work, to the best of the authors’ knowledge, is the first to define and investigate the effect of multiple larger domain shifts that are typically experienced in the AM space, with a demonstration on DED processes. A range of 83 different existing SOTA supervised ML classification models are trained and validated on a range of different machine vision training and validation dataset splits to simulate the different domain shifts. We study the effect of process dynamics from different classes of DED, a broad spectrum of process parameters, and different sensor/AM hardware configurations on the classification performance based on observing melt-pool phenomena.

As such, we introduce the most diverse DED classification dataset to date, to our knowledge. This dataset consists out 9,806 annotated images, iteratively selected using active learning [145] from 285 in-situ videos, acquired from 49 different DED machines. Each video is taken in-situ and represents real-world operation in the field. This is in contrast to the laboratory conditions typically used in the literature [195] [199]. Additionally, different process parameters, camera settings, AM machine settings, hardware positioning, and/or hardware components are used for each video. The result is a highly diverse dataset, which enables splitting the dataset into a training and validation subset in multiple different ways to simulate domain shifts. We analyze and compare the different model architectures by their size, speed (CPU/GPU) and their classification performance across 3 novel domain shift categories, and the naive baseline typically used by the literature (uniform random sampling), for AM, and DED specifically.

Our results show that, dependent on the severity of the domain shift, classification performance can drop by 43.2% relative to uniform random sampling. Based on these in-depth analyses, our findings indicate that the existing literature on ML-based DED defect detection models could therefore have questionable usability as these models are trained and validated on small, non-diverse datasets [195]. Since uniform random sampling is used, which represents the smallest possible domain shift, their performance figures are only representative for a very small operating window, and experienced dynamics at the time of data acquisition. Recommendations on appropriate model architectures and best practices are provided.

4.3 Methods

In this work we study the effect of input data distribution shifts, or domain-shifts, in vision-based data acquired from DED processes, on the performance of SOTA ML classification models typically used for DED defect detection, classification, and/or prediction [195]. This section is divided in 4 subsections. The DED process quality prediction methods are discussed in Section 4.3.1, where the classification task and model architectures are discussed. Our novel dataset, created with diversity in mind, that allows multi-domain validation for DED processes, is discussed in Section 4.3.2. Multi-domain validation methods are presented in Section 4.3.3, where we introduce our novel dataset and novel domain shift simulation methods for DED. Lastly, the experimental details are detailed in Section 4.3.4, where we present details regarding the hardware, software and model training/validation.

4.3.1 DED process quality prediction

The machine learning model details are detailed herein. At the start, the DED process quality classification task is described, followed by a description of general ML classification architecture used, and lastly, a description of the different architectural implementations explored during our experiments.

Classification task

A representation of a typical set of images acquired during in-situ video recording of wire-arc DED processes is presented in Figure 4.1, alongside a close-up view of typical objects present in the acquired images, specifically the torch, wire, spatter, arc, smoke and melt-pool. The classification task used throughout this work is equivalent to the task described by van Houtum et al. [145] where each image is classified separately by its usability for further information extraction. Melt-pool visibility is of special interest since its geometric features are often used and are crucial for proper defect detection as described by the former and other related works [200]. The visibility of the melt-pool is however often obstructed, or occluded, by the other objects making melt-pool related information extraction impossible. Also, sensor related issues such as pixel saturation, an out-of-focus lens, and badly chosen camera parameters such as shutter time can lead to the inability of further information retrieval.

The duration of video recordings is typically longer than the period where the process is active to ensure completeness of acquisition, or due to dwell time in deposition during production. As a result, images without any process are present. Being able to identify whether a process is active/visible could be useful as it provides feedback on whether the process is running or not, as well as feedback on any process interruptions such as lack of material fed into the substrate. Therefore, each image can be classified as either (i) no process, (ii) bad process, or (iii) good process. Examples of each classification as well as the reasons to classify are provided in Figure 4.1. Further details are available in the work by van Houtum and Vlasea [145].

ML architecture for DED image quality classification

Deep learning and convolutional neural networks (CNN) specifically have become a popular tool for machine learning classifications methods. Since 2013, the state-of-the-art (SOTA) in image classification has, with the introduction of AlexNet [201] and the more recent BASIC-L [202], predominately been topped by CNN-based architectures on the Imagenet database [190]. The AM community has picked up on this popularity, as many defect classification methods are based on CNN architectures [195].

Using TF by pre-training CNN backbones on the Imagenet database and fine-tuning a novel classification head, which is tailored specifically to the new task at hand, has proven to be a successful approach. Training time is reduced due to faster convergence, while leading to better performing models compared to training the complete model from scratch [203]. This approach is also favourable for smaller task-specific datasets, as training the classification head only is much less prone to over-fitting than training the complete model due to the large amount of parameters in the backbones. We therefore adopted this approach for our DED process quality classification task.

The output of a CNN backbone, without classification head, is a feature vector of size C , typically different for each backbone. For each CNN backbone used in our experiments, we employ a different trainable classification head which consists of a single fully connected layer which outputs a vector of size 3. A softmax function turns the vector into a probability vector, where each entry represents a probability. The argmax function is used to select the category with the highest probability as the prediction, which is one of the classes as discussed in Section 4.3.1. Before the images enter the CNN backbone, they are resized and tiled to a size of $224 \times 224 \times 3$ using bi-linear down-sampling. This step is necessary to ensure a common input size to the backbone for all images sizes. Image pre-processing is included in the CNN backbones where the 8-bit integer images with a range of 0-255

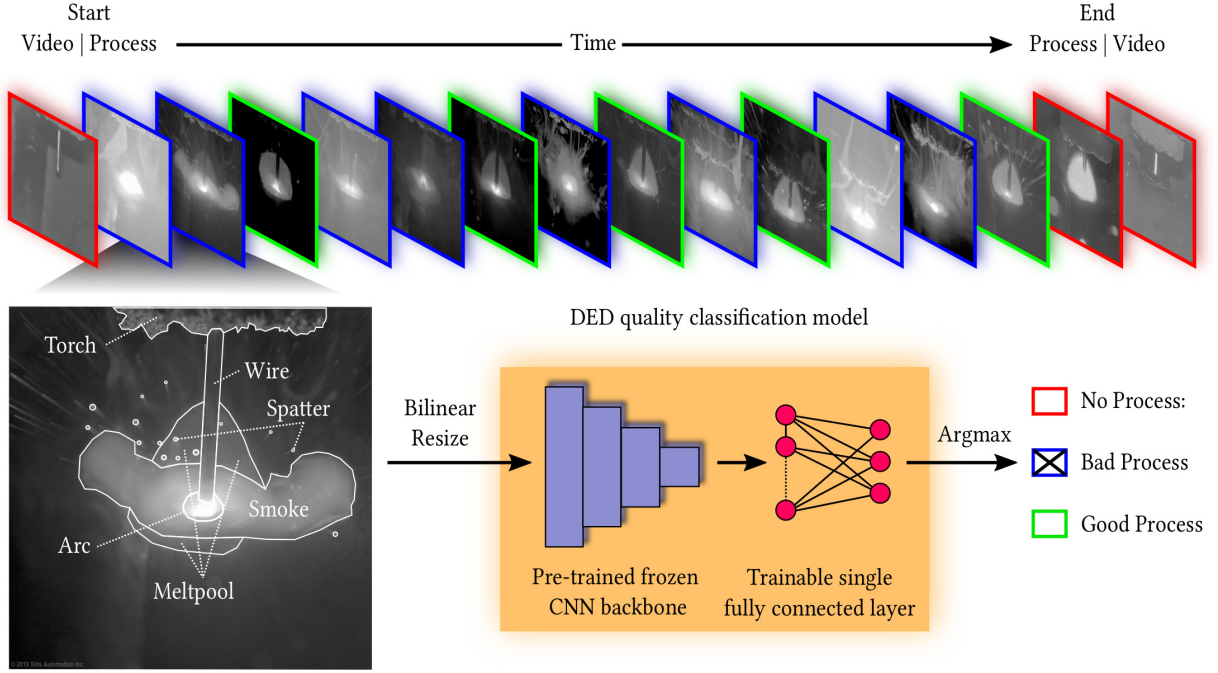


Figure 4.1: A representative example of images from in-situ DED video recordings, an image with annotated phenomena typically observed in DED images, and the general DED quality prediction classification architecture is shown. Image quality categories are represented by color. No process (red); DED process did not start yet, is interrupted, has ended or is outside of the field-of-view. Bad process (blue); DED process is inside field-of-view but not visible due to occlusions caused by any combination of observed phenomena (see bottom left picture). Good process (green); DED process is inside the camera field-of-view and the melt-pool boundary is visible and segmentable. The CNN backbone is pre-trained on the Imagenet database [190] and fixed during training while the classification head is trained to adapt to the new DED defect detection task.

are casted to 32-bit floating point and normalized. A visual overview of the general model architecture is shown in Figure 4.1.

CNN backbone selection

A wide variety of CNN backbones exists to date and 83 Imagenet pre-trained CNN backbones from 10 model families; ResNet(13), RegNet(24), NasNet(2), MobileNet(16), EfficientNet(15), VGG(2), Densenet(3), ConvNext(5), Inception(2), Xception(1) are

commonly available in the popular machine learning framework Tensorflow [204]. Throughout the years, several architectures have been developed with different purposes in mind. Large and complex architectures are generally slow, but provide maximum Imagenet classification accuracy, while smaller architectures enable real-time operation for mobile or edge devices at the cost of accuracy [205]. In general, graphical processing units (GPUs) provide superior performance for model training and inference relative to central processing units (CPU), they are however expensive and are often impractical on mobile or edge devices with limited form factors. The relative difference in model inference speed on a central processing unit (CPU) or graphical processing unit (GPU) can differ as well due to architecture specific implementations.

Combined with a new classification head, as discussed in Section 4.3.1, we train/fine-tune and validate all 83 different classification architectures provided by the Tensorflow framework [204]. The existing classification heads, which are tailored for Imagenet classification, are omitted and replaced with our own custom classification head, as discussed in Section 4.3.1. The total amount of parameters, of the created classification models including our trainable classification head, ranges from 219,315 (MobileNet 0.25) to 348,154,115 (ConvNeXtXLarge) parameters. The number of parameters for the classification heads range from 771 (Mobilenet 0.25) to 12,099 (NASNetLarge). The difference in classification head parameters is driven by the size C of the CNN backbone output feature vector, which differs among the backbones. Additional details regarding the model parameters for all the classification models can be found in the Supplementary material Figure C.3.

Although performance and latency benchmarks have been studied for the Imagenet classification head [205], there does not exist a thorough investigation of the performance of these models specifically tailored for DED processes. Also, the changed classification head results in a different number of parameters, and therefore a different inference speed. We therefore conduct multiple performance and latency tests for each specific model. Specifics regarding performance evaluation methods are discussed in Section 4.3.3.

Model inference latency requirements for AM

Metal AM processes, specifically laser powder bed fusion (LPBF) and DED, differ in deposition spatial resolution (melt-pool size) and temporal resolution (scanning or deposition speed) respectively. Detecting defects in-situ requires the ability to observe every scanning/deposition location during the build with high sampling fidelity matching the spatial and temporal resolution of the specific process, otherwise important defects could be missed [206]. If the field-of-view (FOV) of the detector is small, a high

frame-rate is required to ensure the energy source-material interaction phenomena are captured with good spatial and temporal resolution. A large FOV would cover a larger area, therefore reducing the required frame-rate to be able to cover ever spatial location; however the reduced information density (temporal resolution) could lead to the inability to observe important dynamics and micro-structural defects. A small FOV might observe local dynamics only, and potentially miss global dynamics that contribute to the formation of defects.

In-situ observations of the melt-pool geometry have been proven to be a popular and effective data stream used to predict the occurrence of defects [145, 200]. The minimal required FOV of the camera that is able to observe the complete melt-pool could be a sufficient starting point, providing a lower boundary on the required camera frame-rate. The melt-pool size is however dependent on many factors, including the material feed-stock rate, energy-source power and velocity. The energy-source spot size, or beam spot diameter, could be a better constraint as it is indicative of the smallest possible melt-pool size. Beam-spot sizes typically range from 1.0-5.0 mm and 0.05-0.2 mm for DED and LPBF respectively, while scanning speeds lie between 5-70 mm/s and 500-3000 mm/s [207] respectively. A lower bound can be calculated based on these metrics to ensure that every location is captured, with sampling happening at least once per melt-pool length. DED would require a minimal frame-rate, in frames per second (FPS), of $70/1.0 = 70$ FPS. LPBF on the other hand requires much higher frame rates at $3000/0.05 = 60,000$ FPS.

The time between the acquisition of two frames, or frame-rate, is the maximum time the quality classification system is allowed to consume. Besides quality prediction time determined by the ML model architecture complexity, communication between the camera, model and AM system is required as well. A frame-rate in the 100+ FPS range is what we aim for in this study on DED systems to leave room for communication and other types of computation overhead. We use this as a constraint in our model architecture selection process as discussed in Section 4.4.

4.3.2 Dataset

The novel DED dataset considered for this work has a broad diversity in scenery. The dataset is a continuation, or super-set, of the dataset introduced in [145]. A set of 285 wire-arc in-situ videos were acquired from 49 different DED machines. Each video is taken by either a Xiris XVC-1100 or a XVC-700 NIR camera. These cameras are specifically tailored for welding or AM applications, as they feature a high dynamic range (+140dB) and are able to handle large differences in brightness. These cameras are monochrome,

producing a single channel gray-scale output resolution of 1080x1280x1 at a frame-rate up to 55 frames-per-second for the full resolution. Higher frame-rates can be achieved by reducing the camera field-of-view setting proportionally. A total number of 557,164 images were acquired; this data set was deemed too large to be annotated completely. To visualize the differences, we have selected an example image for each machine/scene in the Supplementary material Figure C.1.

Furthermore, in terms of data selection for annotation, it is important to note that in industrial settings, similar images are present due to static recordings before or after the deposition process or due to steady state process dynamics and tool paths; this leads to redundancy in annotation efforts, as discussed in Section 4.3.1. For this reason the AWUS active learning method was deployed [145], to iteratively select annotation instances for 5 domain-expert annotators to annotate 9,806 images iteratively. This increased the chances of having a smaller, yet diverse dataset. The annotation protocol used to link each image with a specific class is provided in Figure A.1 of Supplementary material A.

Statistics of the resulting dataset after annotation are provided in Table 4.3.2. The relative class balance between the no-, occluded-, and segmentable-meltpool are 37.5%, 17.1%, and 45.4% respectively. Each specific hardware setup, consisting of a specific DED machine, with a specific camera setup (machine position and rotation, optical lens, etc.) leads to a certain visual scenery. We therefore refer to these specific setups as "scenes" from now on. Table 4.3.2 provides insights into the distribution of images, videos and classes across the scenes. Each scene has at minimum 31 annotated images, and 1028 at maximum.

Most videos and scenes contain labels from all 3 classes, some however contain only two or a single class. During data acquisition, recording generally started before the process started. However, in certain videos, this is reversed, leading to less no-meltpool class images, and in some cases, none. Other videos, observe DED processes tuned with either very good or bad process parameters, therefore leading to either many occluded or segmentable class images. Experimental details on training and validation splits is further discussed in Section 4.3.4.

4.3.3 Multi-domain validation

ML model performance will never be completely predictable for unknown input-data. Training ML models and validating them on data from a different input-data distribution might provide insights into what real-world performance can be expected when deployed under different operating conditions that share similarity with the validated domain

Table 4.1: Xiris dataset statistics

	Total	Per Scene			Per Video		
		Min	Mean	Max	Min	Mean	Max
Scenes	49	-	-	-	-	-	-
Videos	285	1	5.8	34	-	-	-
Images	9806	31	200.1	1028	1	34.4	135
Classes	3	1	2.9	3	1	2.6	3
-None	3685	0	75.2	765	0	12.9	118
-Occluded	1666	0	34.0	165	0	5.8	45
-Segmentable	4455	1	90.9	687	0	15.6	124

shifts during the training process. Validating ML models on different domains has been studied before [105], but has yet to be investigated within AM. Our approach is therefore, as far as we know, the first. Simulating domain shifts requires a highly diverse dataset that can be split into a training set and multiple validation sets, where each validation set represents a specific domain shift type.

There is a literature gap pertaining to specific definitions or exhaustive lists of possible domain shifts within AM, and specifically DED. Furthermore, there is a gap in DED machine learning literature (and AM machine learning literature in general) that focuses on domain shift simulated through specific data splitting methodologies. These gaps pose a significant impediment towards reliable adoption of lab-scale ML models to industrial or real-world applications. In the present work, we therefore propose a novel, and to our knowledge first, domain-shift simulation standard for DED that can be used for ML model training and evaluation. This standard is based on the different domain shifts identified in our novel dataset (described in Section 4.3.2). The work stands to help the AM community for ML model performance comparison and provide insight into model performance under different domain-shifts possible within DED.

Domain shift - hardware shift

Each of the 49 different DED machines has been used to take a single or multiple videos of the deposition process using a single camera (Xiris XVC-1100 or XVC-700), mounted either co-axially or off-axis. The hardware setups for each machine remained fixed for the duration of each individual experiment conducted. The effect of the differences in hardware in the visuals captured by the camera is the first proposed domain shift to be investigated.

A ML model that is able to perform well under a change in hardware or positioning could be a powerful tool, as it would scale to many different use cases and different machines. Simulating this domain shift can be achieved by dividing the data based upon the DED machine used. After investigation of the dataset, we observed the following hardware-related differences which are kept constant across the multiple experiments conducted on each specific machine: (1) the camera is placed at either a leading, trailing or side-view position; (2) camera zoom is different, as different optical lens setups are used; (3) the camera is rotated in different ways relative to the DED machine; (4) the camera is positioned at different distances from the process zone; (5) the camera is either fixed to the DED machine (moving at the same rate as the deposition head) or to the build plate (stationary with respect to the deposition head); (6) tool-path patterns are either straight lines, or zig-zag patterns. An example of a change in scenery can be observed in Figure 4.2 under validation set D.

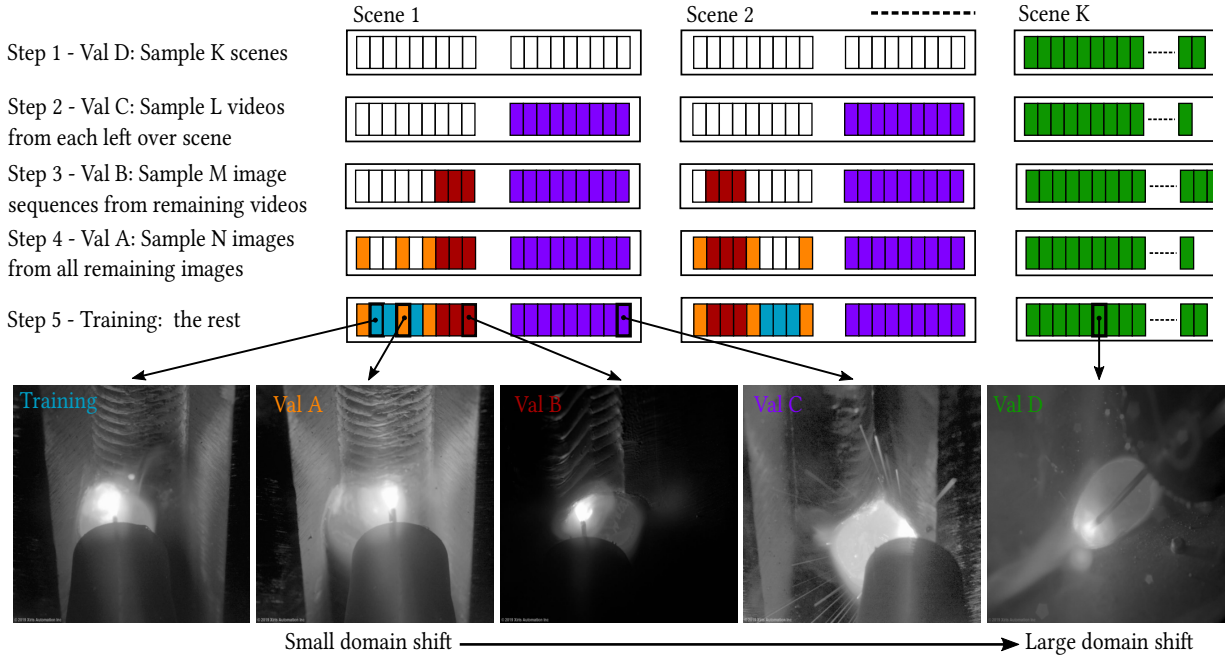


Figure 4.2: Validation dataset differences and examples. Images in the training dataset and validation dataset A are sampled at uniform random from the videos. Validation dataset B are image sequences taken out of each video at uniform random. Validation set C are images from videos where the process is using different process parameters and validation D are images from videos acquired while observing different processes.

Domain shift - parameter shift

Each video within a scene is considered a different experiment, as different operational settings are used for each trial. Machine settings such as feed-stock material type, and process parameters differ between experiments, leading to different visuals, as phenomena such as smoke and spatter (see Figure 4.1) can be excessive under sub-optimal operation. In turn, these phenomena can influence the camera operation since pixel saturation, spatter, smoke and other occluding features can lead to bad process visuals, as discussed in Section 4.3.1. Training a model on a range of process settings and parameters, and validating on other settings and parameters, but conducted on an equivalent hardware setup (or scene) is what we define as a parameter domain shift. Companies typically do not have many different DED machines at their disposal and could only be interested in a ML model that performs well under any process settings or parameters. For this work, there are however many unknowns, as we do not have access to the specific settings and parameters used. The only certainty is that these videos are generated on different machines, where each machine is tuned with its specific process parameters. Additionally, none of the scenes have influenced each other. The videos within a scene differ from one another typically in the following ways: (1) different tool-paths with a single, multiple, or overlapping deposition lines; (2) differences in spatter, smoke, melt-pool size and shape caused by changes in process parameters; (3) geometrical build plate differences caused by previous depositions, or build plate shape differences such as flat plate versus corners. All 285 videos have at least some form of these differences present; some smaller, some larger. An example image of a process under different process parameters can be observed in Figure 4.2 under validation set C.

Domain shift - time shift

Every video consists of a sequence of images. The dynamics of the process might however change in time as heat accumulation could affect the process signatures. The tool-path might also affect the process indirectly, as heat affected zones might be visited more than once and geometric defects in the substrate may influence the instantaneous process zone. Training and validating a model on different sequences of images within each video might provide insights into the effect of these changes in process dynamics on the performance of the ML model. The flexibility of AM and nearly limitless freedom in fabrication has the potential to fabricate different parts without much effort. These parts will however be executed based on different tool-paths and resulting process dynamics. New parts will therefore always lead to unknowns in terms of quality and observable process dynamics. We therefore consider this domain shift of type B.

Domain shift - minimal shift

A minimal shift is what we consider the smallest possible shift that can be generated using a specific dataset. This type of shift is achieved by not taking into account any of the former and simply select data-points at uniform random for training and validation from the same experiment set. This has been the choice of all literature on defect detection in DED. The insight this specific type of domain shift provides is limited in the sense that the only differences between the training and validation dataset distribution is based upon local dynamics, which are specific to each image are not present in other frames. This type of validation is specific to the DED machine, process settings/parameters, camera settings/parameters, tool-path, build-plate, and every other specifics used during data acquisition. We designate this a minimal domain shift as type A. An example can be observed in Figure 4.2.

4.3.4 Experimental details

This section provides details regarding the domain shift simulation methods and model training/validation implementations.

Simulating domain shifts

In this work, we simulate/validate the performance of ML models during training on the 4 different domain-shifts mentioned previously. The total dataset of 285 videos is therefore divided/split into 5 sub-datasets. A single training and 4 validation datasets (Val A,B,C and D). There are however multiple ways of splitting the data and a lucky or unlucky split could provide an unrealistic result. Training and validating on all possible combinations of training and validation sets is not practical as the numbers grow very large. We approximate the distribution of possibilities in combinations by conducting 30 different cross-validation trials where each training and validation trial is selected using uniform random sampling.

The algorithm used to create the different dataset splits for each trial is visually represented in Figure 4.2. At first, a set of K scenes is selected at uniform random and all the videos, and therefore images are set apart as Validation dataset D. From each remaining scene, L videos are selected at uniform random at set apart as Validation dataset C. From each remaining video, M image sequences are selected at uniform random and set apart as Validation dataset B. At last, N images are sampled at uniform

random from all remaining videos. All images that remain are combined into the training dataset.

During the selection of each validation split we take the imbalance in images/classes per scene/video into consideration. Scenes that are selected for the training dataset are required to have multiple experimental videos, this is necessary to be able to split the videos, one set of videos are reserved for the training dataset, while other videos are assigned to validation set C, to be able to simulate the process parameter shift. The number of classes present in the training videos should be equal or larger than the videos in validation set C, to ensure every class has been seen before. To be able to split videos into a training part and validation sets A and B the number of annotated images in the video should be sufficient. We selected videos with at least 9 data instances, 3 for each class present in the video.

As discussed in Section 4.3.2, there exists a class imbalance in our dataset. During the creation of the dataset splits we maintain this class imbalance roughly for each split. Each validation dataset represents approximately 16.67% of the complete dataset at each trial. The resulting training datasets for each trial are approximately 2 times larger than each validation dataset at 33.33%.

Model training and validation

Each one of the 83 classification models, as discussed in Sections 4.3.1 and 4.3.1, is trained with the CNN backbone parameters frozen, such that only the classification head parameters are trained. We use a mini-batch size of 32, the Adam optimizer [208] with a fixed learning rate of $1e - 3$, a categorical cross-entropy loss (CCE) function and data augmentation with random flipping in the vertical and horizontal direction during training. We keep track of the model weight using an exponential moving average (EMA) with an decay rate of 0.999 which is updated after each mini-batch. We swap the model weights with those stored by the EMA for model evaluation on the different domain shifts as is discussed in Section 4.3.3, and swap them back with the original weights before training continuation. We use early-stopping during training, meaning that we stop training whenever the CCE loss of the minimal shift, or validation set A, has not decreased 2 epochs. The choice for 2 epochs is estimated empirically, and seems short. The use of EMA resulted however in a very smooth validation curve such that we were able to pinpoint within 2 epochs if the models were over-fitting.

The number of trainable parameters in the classification heads range from 771 to 12,099. Although most smaller backbones have smaller output feature vectors C , there are many

exceptions, as shown in Figure C.3 in the Supplementary material. The relatively small amount of parameters to optimize led to short training times, fast convergence and no need to add additional regularization, such as dropout, as it did not offer any advantage during our tests.

After each epoch, the models are validated on all 4 domain shifts, validation sets A, B, C and D. We record the average loss and accuracy over each epoch for the training and validation datasets for all 30 domain-shift simulation trials as discussed in Section 4.3.4. Other metrics, such as the F1-macro score, could be included in future studies, we however decided that was beyond the scope of this study, with the relative performance between domain-shifts as the main area of focus. Class imbalance is present in our training and validation datasets, and solutions to compensate, such as oversampling and class weighting, during training could remedy this problem. This is however considered in future work, as well for the aforementioned reason. All experiments are conducted on Ubuntu 22.04.2 LTS using Python 3.9 and Tensorflow 2.13 for model training and validation. Hardware includes a Intel i7-9700K CPU and Gigabyte RTX 3090 Turbo GPU.

4.4 Results

This section presents and discusses the results from all training and validation trails. All 83 models were trained and validated on 30 different cross validation trails and their inference speed on the CPU and GPU were evaluated as well. Detailed quantitative performance and inference speed results per specific model can be observed in Supplementary Figures C.2 and C.3 respectively.

4.4.1 Overall results

The overall performance results are presented in Figure 4.3 where all training simulations of the 83 classification architectures combined are shown. As anticipated, validation dataset A is performing best with an average accuracy of 81.8%. Within this domain shift category (Val A), the worst performing model run achieved an accuracy of 75.9% while to best model run reached 86.6%. As validation dataset Val A represents the smallest domain shift, it confirms our hypotheses that small domain shifts lead to higher performance. This is corroborated by similar performance reported in literature [4, 144], where this is the most common domain shift. If we compare Val A with Val B, Val C and Val D, the larger the domain shift, with Val D being the largest, the lower the classification performance, with

the worst model run reaching an accuracy of 35.0% on Val D. Model performance results for the different validation sets relative to validation set A are presented in the subplot on the right of Figure 4.3. The average performance loss for Val B is -2.5%, Val C is -5.8% and val D with -20.8%. We observe that model performance can drop by -43.4%, -16.2% and -10.1% in accuracy relative to Val A whenever operated in environments similar to Val D, C and B respectively. This drop in performance is not uncommon and has been studied extensively in areas such as medical image segmentation [209].

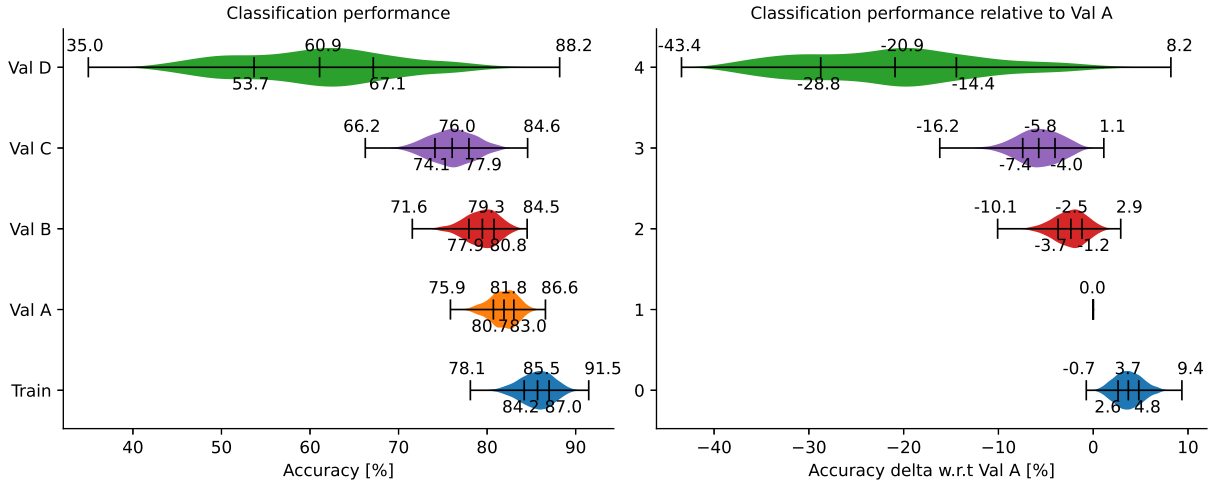


Figure 4.3: Overall classification results, collected from 30 cross-validation trails for each of the 83 different classification architectures. Each dataset, excluding training, represents a different domain shift. Val D (hardware shift) is the largest domain shift followed by Val C (parameter shift), Val B (time shift) and Val A (uniform random sampling). The left subplot provides insights into the accuracy while the right subplot shows relative performance towards the general case, uniform random sampling, Val A.

An interesting observation is the spread in the performance results for each validation dataset, which is made visible by the violin distributions in Figure 4.3. As expected, the spread in the performance results increases with domain shift severity as the results for Val D show significantly larger variation compared to Val A. The increasing spread can potentially be explained by the variation in validation datasets between trails. Each trail selects a different subset of videos from different processes out of the 49 at uniform random to create a different validation set D. As there is expected to be a significant variability between the images in the videos from each process, there is also a larger difference between the validation datasets in Val D. Although not as severe, this is also expected to explain

the increasing spread in validation datasets A, B and C. This work demonstrates that the transitional CNN type image classification models are able to generalize to certain changes in the environment rather well, such as Val A, B and C. The results on Val D do however imply the need for different types of ML models that generalize in performance across changes in hardware if the trained model is expected to generalize to different DED machines.

4.4.2 Relative performance

As uniform random sampling is often used for validation in the literature (see Section 4.2), we investigate how the performance on Validation set A (uniform random sampling) compares to the performance of each model architecture on the other validation datasets. If a good performing model on validation dataset A, performs well on other domain shifts, it might provide insight into what model performance to expect for other domain shifts while only validating model performance under uniform random sampling, such as Validation set A. In many cases for AM, the dataset available for the development of an ML model is too small to simulate different domain shifts, as is discussed in Section 4.2. If an initial performance evaluation using uniform random sampling could provide an initial guess on what the performance will be under different domain shifts, it might help the AM community in developing better and more practically oriented ML solutions, specifically solutions in the in-situ defect detection space.

An overview of these results are presented in the bubble chart Figure 4.4 where the size of the bubbles represent to model architecture size in terms of total amount of parameters. Model architectures that in general perform well on Val A, perform well on the other datasets. There is however a significant performance drop depending on the domain shift as previously discussed in Section 4.4.1.

The best performing models across the board of validation datasets generally have a backbone from the RegNet family with RegNetY320, model number 74, performing best on validation datasets A and B, see Figure 4.4(a), over all cross-validation trails. RegNetY160, number 70, slightly outperforms RegNetY320 on Validation dataset C, Figure 4.4(b), while MobileNetV3Large_1,0M, with number 5, performs best on validation dataset D as shown in Figure 4.4(c).

Another observation of all sub-figures in Figure 4.4 is that larger models, represented by the larger bubbles, in terms of number of parameters tend to outperform smaller models. There are however exceptions, such as MobileNetV3Large_1,0M which is small in size but performing very well, especially when compared against large models such as

ConvNextXLarge, number 82. Model architectures with a lower amount of parameters tend to be faster in inference on the CPU and GPU [205]. Although performance is important, the trade-off between speed and performance might be crucial as a slower model might not be fast enough. For DED specifically, a slow model architecture might not be able to predict in real-time, and not be able to observe important dynamics under motion.

4.4.3 Performance/Speed trade-off

Sufficient inference speed of the model architecture is crucial for the application of real-time defect detection in DED or LPBF, as discussed in Section 4.3.1; typical DED systems require ML model architectures capable of running at 100+ FPS. We therefore conducted 30 repeated model inference speed experiments for each of the 83 models on the CPU as well as the GPU. Figure 4.4 (d,e) show the average validation accuracy over all validation sets A,B,C and D versus the CPU and GPU speed respectively for each of the 83 classification architectures. The size of the bubbles represents the total number of parameters in the model.

It becomes apparent that the model family most appropriate for DED, and possibly LPBF purposes, are the MobileNet architectures, as they all reach 100+ FPS on the CPU. The ConvNeXt family are outliers compared to the rest, which could be related to the specific software implementation. The additional parallelization a GPU offers. compared to a CPU, leads to higher speeds, easily surpassing 100+ FPS for most models except very large models such as ResNet350, number 76, and ResNet420, number 77, are relatively slow.

Models that reside at the top right corner of both plots provide the best balance in terms of accuracy versus speed. MobileNetV3Large_1.0M is standing out as it is fast on both the CPU and GPU, while providing good accuracy. Models that perform well on the CPU, perform well on the GPU in general and have a low amount of parameters. The comparison between model inference performance on different hardware architectures has become a popular topic with the introduction of edge devices such as studied by Suzen et al. [210] and might serve the AM community for future ML studies in the selection of CNN backbones.

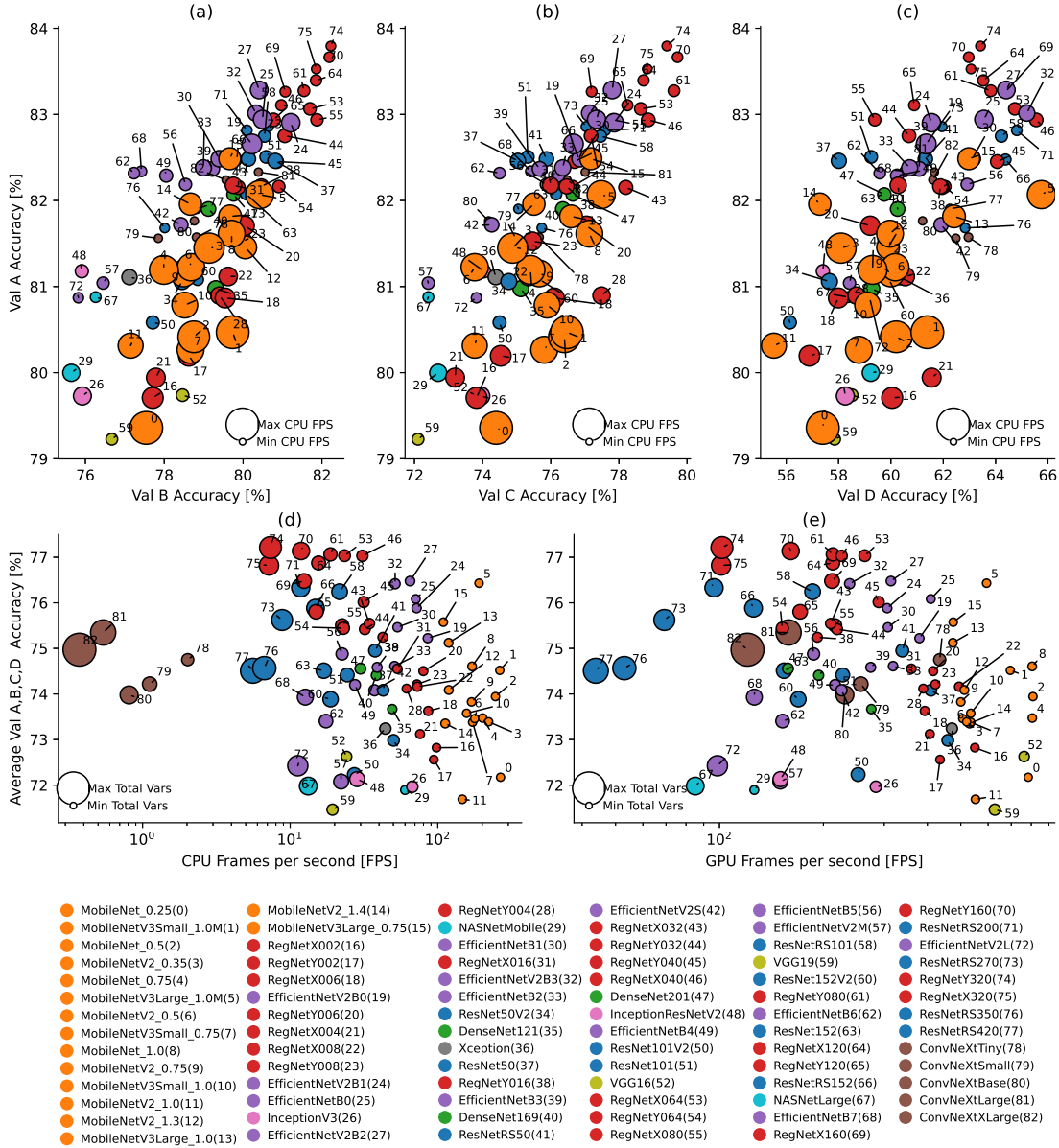


Figure 4.4: (a,b,c) Relative performance between architectures and datasets. Colors represent model family. Bubble sizes represent relative CPU speed. (d,e) Average validation dataset performance against CPU and GPU inference speed. Bubble size represent model parameters.

4.5 Discussion

When comparing the findings herein with the findings from literature, similarities can be observed in terms of concluding that if domain shifts exist from the training data, this generally leads to worse model performance. The authors of [105] have found that a simple pose change, in the form of rotation, scaling, or any other geometric transformation, of familiar objects will lead to a drastic decrease in classification performance, as 97 % of the objects were mis-classified. Their findings are conducted on the Imagenet Database using SOTA CNN models, including a AlexNet, ResNet, and Inception architectures. The consequences of their results can be severe, since objects in operation (in real life), are very likely to appear in a different pose. The question that therefore arises is whether these models trained on Imagenet are useful at all beyond their training dataset. Similarly, for DED and AM in general, changes in operating conditions due to hardware changes, process parameter changes, materials, part design changes, toolpath changes or other external factors like lighting might result in model performance degradation. The current work showcases a worst case performance drop of 43 % whenever the hardware, or DED machinery, is changed.

Another more comprehensive analysis of model performance under data distribution shift was conducted by Wiles et al [211]. They trained and validated more than 85,000 different model architectures using a variety of datasets, and data distribution shifts. They propose that the development of a new model for every specific domain is impractical, and instead the focus should move towards domain adaptive models. Furthermore, they find that pre-training models on existing datasets such as Imagenet, lead to better performance under domain shifts. Finally, they state that it is crucial to validate models using different domain shifts to understand realistic model performance in practice. Therefore, an important research direction for machine learning in AM, particularly for DED, is to explore adaptive models and human-in-the-loop models that can be tailored to new domain shifts with minimal annotation and re-training, such as the work on active learning by van Houtum et al. [145].

The results presented in this present work show similar behaviour as described by [105] and [211]. Therefore, we bring into question the usability of the existing machine learning model architectures beyond their training dataset in AM, as reviewed by [195], as they are trained on excessively small and non-diversified datasets, and furthermore validated on data from the same distribution. The findings in this present work show that fairly common domain shifts in the AM space, as discussion in Section 4.3.3, lead to a drastic drop in performance.

In the authors' opinion, future work in defect detection and any other machine

learning application in AM should aim to focus more on model robustness under different (but plausible) operating conditions in practice, instead of maximum model performance under ideal and un-realistic operating conditions. Furthermore, if datasets for such robust performance testing criteria are not fully available, there is a need for changing the direction towards adaptive models such as the work by Oh et al [110].

4.6 Conclusions

This work presents a machine learning model validation methodology based on the simulation of different domain shifts. The findings were demonstrated by introducing a novel diverse dataset consisting out of 9,806 images selected and annotated from set of 285 videos, generated by observing 49 different DED machines in real-world operating conditions. The multi-domain validation strategy exposes machine learning model performance when used in different environments and can help the additive manufacturing community with the development of multi-domain models which perform better under changing environments. The contributions and conclusions are as follows:

1. The work defines a standard set of domain shift categories of interest for DED (and AM in general) that can be used to develop more robust ML models for the AM space that provide practical performance insights.
2. The work illustrates that current ML architectures are performing relatively well, if the domain shift is relatively small, i.e. small process parameter changes, unseen dynamics and tool-path changes, with an average performance drop between -2.5 and -5.8 % compared to uniform random sampling of the same domain.
3. The work illustrates that hardware changes, such as changes in sensor type or machine type lead to large performance drops with an average of -20.9 % in classification accuracy.
4. The work provides an overview of the applicability of 83 CNN backbones for DED quality prediction with regards to classification performance and inference latency.

Chapter 5

AMVOS: Semi-supervised video object segmentation for additive manufacturing¹

5.1 Preface

Additive manufacturing (AM) involves intricate processes with a multitude of process parameters and operating variables that typically fluctuate either stochastically or deterministically, or vary between machinery, processes, builds, and materials to optimize part quality. Real-time (in-situ) process measurement using machine learning (ML) has emerged as a promising approach to enhance quality control, yet it remains challenging in the face of varying process conditions. Current ML methodologies primarily focus on constructing models tailored to a narrow operational window of a specific AM machine, which typically requires model retraining on newly acquired and annotated data from the target operational environment. While methods like transfer learning, domain adaptation, and active learning have been proposed to reduce retraining time and resources to create new ML models, they still require them. In this work, we introduce a novel training-free domain adaptation ML framework capable of adapting to diverse operational contexts requiring image/video segmentation. This innovation leverages human-in-the-loop interactive annotation, guided by insights from the architecture itself. Notably, the proposed architecture demonstrates real-time processing capabilities, achieving state-of-the-art inference latency at 155 frames per second (FPS). Performance

¹Chapter is adapted from an article submitted for publication

evaluation is conducted on vision-based in-situ directed energy deposition (DED) melt-pool segmentation, and indicates that merely a few annotations (less than five) are needed to attain an impressive 83+% intersection-over-union score.

5.2 Introduction

Additive manufacturing (AM) offers compelling advantages over traditional methods, such as enhanced design flexibility, customizable and complex parts, and lower tooling costs—all achieved through material waste reduction and simplified supply chains [10]. Quality control is however challenging due to the variability in design, material, operating conditions and/or AM process/machine variations. Re-tuning of process parameters is typical for each variation in experimental conditions; this is expensive and time-consuming, but necessary to achieve consistent part quality [21]. Although these challenges may restrict the benefits and adoption rate of additive manufacturing (AM), manufacturing variations offer valuable insights into part quality and process performance across different operational settings. Analyzing data from part design, material characteristics, in-process dynamics, and post-process attributes can enhance the AM process [212]. Advancements in sensor technologies and data acquisition methods have greatly expanded the amount of AM data available before, during, and after part fabrication, creating new opportunities for knowledge discovery and quality enhancement. Recent efforts to improve process repeatability and part quality focus on gaining comprehensive understanding through in-process monitoring and analytics [197]. Machine learning (ML) has become a popular choice to leverage these data streams and generate models that are able to provide feedback on process and part quality during and after fabrication and report on detect defects throughout the entire AM development cycle [194].

Directed energy deposition (DED) is a type of metal AM process where a laser melts a feed-stock material delivered onto a substrate to form parts or features. DED processes yield a significant amount of in-process data that can be harnessed across various scales of resolution, including spatial, temporal, spectral, and radiometric dimensions [213]. Vision-based in-process observation of the molten material, called melt-pool, created through the interaction between the energy source and feed-stock material has been studied extensively and forms a popular data stream for the creation of ML models for real-time in-situ control in DED [214]. Image segmentation models are particular popular in DED as they provide feedback that can be utilized to develop closed-loop feedback control algorithms that control the geometrical dimensions, leading to more accurate

deposition and part dimensions [215].

The performance of ML models in AM applications has been studied by Fu et al. [195], and model prediction accuracies typically exceed 90%. The data used to create and validate these models is however typically small (<1000) and low in diversity (single AM process/machine, small process parameter set). Large-scale machine learning systems for industrial applications, such as AM and DED, suffer from a lack in adaptability and scalability as reported by Lwakatare et al. [216]. Naive ML implementations and evaluations, such as the models studied by Fu et al. [195], tend to report performance only on a very specific setup, without investigations on model performance caused by operational variations or environments. These models therefore lack in their ability to adapt and scale, which typically is of importance in industrial settings. AM machines are complex machines, with a vast number of possibilities in combinations of process parameters, machine settings, material types, part designs, and tool-paths. Experiments are typically expensive and time-consuming as well. The creation of sufficiently large and diverse datasets in AM is therefore currently impossible leaving only room for a limited amount of AM experiments to inform and validate models [198].

Transfer learning (TL) is a type of ML where pre-trained models are re-used for a new task, typically leading to better model performance, and requiring less target training data. Pre-trained convolutional neural network (CNN) backbones on the Imagenet classification dataset [190] are a popular transfer learning choice in AM, as these backbones operate as general image feature extractors, for any image input distribution, and can be re-trained for different tasks, such as DED image segmentation [215]. Domain adaptation (DA) is a type of TL where the source and target data have the same task, but different input data distributions. In DED melt-pool image segmentation this could translate to developing a model on data acquired from DED machine A and using DA methods to re-train and adapt the model for operation on DED machine B using a much smaller dataset acquired from in-situ process observation of DED machine B [199]. Target domain data is required for re-training based TL and DA methods, but often not practical in AM and DED, where time-consuming and expensive experiments would be needed every time a change is made to the input distribution domain. Any change made to the AM process could require a new re-trained model, which is orthogonal to the manufacturing flexibility of AM machines [198].

Video object segmentation (VOS) is a relatively new type of ML aimed at segmenting and tracking objects in image sequences. Semi-supervised VOS (SVOS) refers to the setting where a few initial hints, in the form image and object mask pairs are provided, and the other masks of that specific object should be predicted for the remaining images in the sequence. Interactive VOS (IVOS) extends the SVOS setting by providing human-in-the-

loop interaction through scribbles. From SVOS predictions, a select few is annotated by humans, combined with the image, and added to the stack of initial hints [217]. SVOS, and even more so IVOS, are extremely challenging tasks, as it asks objects to be segmented while experiencing changes in occlusions, motion, appearance, and perspective. Furthermore, the model should be object agnostic, meaning that any type of image and object could be segmented and tracked without any model re-training. Additionally for IVOS, a selection algorithm is required to provide the interaction. Memory-based matching methods are VOS architectures that have gained particular attention since the solution provided by Oh et al. [110] has been proposed. These methods [218, 219, 220, 221] form the state-of-the-art in VOS and employ a memory bank with encoded features from previous images, annotations and predictions. Encoded features from the current image are used to match and select specific memory features, combined and decoded into the a prediction mask. The difference between TL, and therefore DA, and VOS methods is four-fold: (1) TL typically requires model re-training to adapt, VOS does not; (2) TL requires relatively large amounts of annotated data, VOS requires few; (3) TL in AM segmentation methods are object specific, VOS is object agnostic, and (4) TL methods have been explored in AM, while VOS methods have not to the best of our knowledge.

This work is the first to investigate the applicability of VOS methods in additive manufacturing (AM), specifically in DED melt-pool segmentation. The main five contributions are summarized as follows: (1) A novel scalable VOS architecture is proposed, featuring a simplified cross attention and decoder design, as well as a single CNN backbone encoder, making it faster than the current state-of-the-art; (2) A novel object-specific and object-agnostic training regime is introduced, enabling VOS architectures to be trained for specific objects, a concept previously unexplored in the VOS literature; (3) The first application of VOS in AM is demonstrated through a case study on DED melt-pool segmentation; (4) A series of performance and run-time comparisons is conducted, examining the effects of object-specific versus object-agnostic training, static and dynamic memory sizes, and pre-training on unrelated datasets; (5) The impact of human-in-the-loop annotation in the DED melt-pool setting is studied, utilizing complete annotation feedback instead of mere scribbles.

Our results show that the novel proposed VOS architecture is able to adapt to newly unseen DED process environments using only a few (< 5) annotated images, reaching an 83+% intersection-over-union (IOU) score. The outcomes indicate that training-free model adaptation in the form of VOS is possible for AM related tasks and that it could potentially solve the limitations of TL and DA. Only a few annotations are required for VOS to adapt to a change in process, parameters, settings or any other change made to the experimental AM environment. It could prove to be pivotal in the development of ML methods in the

field of AM where flexibility and on-demand manufacturing is key.

5.3 Methods

In this section, a general description of video object segmentation (VOS) is provided. Thereafter, the focus will be to introduce a novel VOS architecture, called AMVOS, specifically tailored for AM and DED processes aimed at fast inference and architectural simplicity. Thirdly, the different operational modes of AMVOS are discussed, followed by a description of the training and validation experiments conducted. At last, implementation details are provided.

5.3.1 General VOS description

VOS architectures generate segmentation mask probabilities P for each provided input image I , for any resolution image. In memory bank based VOS methods [217], initially provided information, called hints, in the form of pairs of ground-truth object masks and corresponding images $[(I_1, M_1), \dots, (I_N, M_N)]$ can be provided and stored. During operation, previously predicted object mask probabilities $[P_{N+1}, \dots, P_K]$ and corresponding images $[I_{N+1}, \dots, I_K]$ are stored alongside the initially provided hints in memory. Future prediction P_{K+1} therefore has access to initially provided ground truth and previous predictions information. See Figure 5.1 for a visual interpretation of the application of VOS on DED melt-pool data and footage of a human riding a bicycle.

State-of-the-art memory bank VOS methods typically use an encoder, decoder, and cross attention module in their architectures [110, 218, 219, 220, 221]. Typically two separate encoders are used, one to encode image information into lower dimension features, the other to encode ground-truth masks, and predicted masks. These encoders are typically part of the ADD and PREDICT functional components in VOS architectures as shown in Figure 5.1. A cross attention module is employed where current frame features are matched with memory features, created from encoding previous images and masks. Based on the feature similarity, typically calculated using dot-product attention followed by a *softmax* function, historical object and image memory features are selected and concatenated with the current image features. The decoder turns the combined features into output probability mask for each objects using a soft aggregation method introduced by Oh et al. [110]. The cross attention and decoder modules are typically part of the PREDICT block as shown in Figure 5.1.

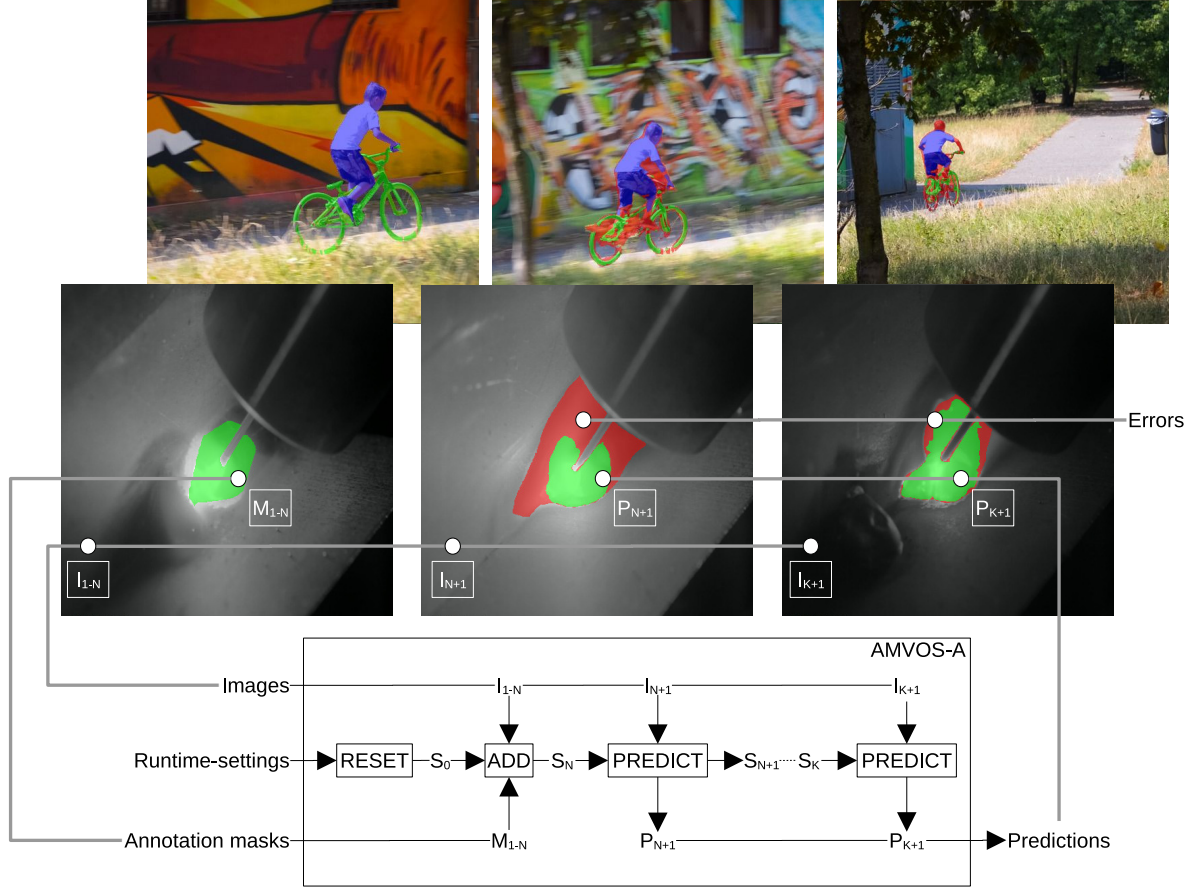


Figure 5.1: AMVOS-A; Semi-supervised video object segmentation (VOS) on directed energy deposition (DED) melt-pool data, and any other data represented by the human an bicycle.

The amount of initial and previous information provided and maintained during VOS operation is considered the memory management policy. In the standard SVOS setting [222], a single initial hint is provided in the form of the first image-object masks pair $[I_1, M_1]$. Several studies [220, 221] indicate that keeping the first initial frame, and the last prediction into memory leads to the best performance-latency trade-off. Other research [110, 218] has focused solely on performance, often employing complex and typically slow memory storage decision-making policies. In contrast, IVOS leverages an oracle, typically a human, to intervene in the VOS process and provide new ground-truth data as initial hints. Specifically, IVOS utilizes predictions generated by a SVOS model to select and

re-annotate a single frame and its prediction masks.

5.3.2 VOS requirements for AM

Although fast VOS methods have been proposed in recent literature [220, 221, 219], with such methods reaching reaching 100+ frames per second (FPS), these methods utilize a first and last only memory policy. A single initial and the previous prediction are stored in memory alone, resulting in small memory occupation, fast cross attention, and therefore fast VOS operation. Keeping such small amounts of information in memory might work for SVOS settings, but does not translate to IVOS aimed at iteratively adding informative new ground truth information. Their results can therefore be seen as an upper-bound to inference performance, while not practical for IVOS purposes.

The size of the laser beam spot sizes in DED typically range from 1.0-5.0 mm while scanning speeds lie between 5-70 mm/s [207]. The beam spot size is a good indicator for the smallest possible melt-pool, and combined with the requirement to capture melt-pool information at every deposition location, an upper bound on the maximum model prediction inference latency can be calculated of 0.014 ms or 70 Frames per second (FPS). Combined with data acquisition latency, we aim for 100+ FPS for our VOS architecture, discussed in Section 5.3.3.

Existing VOS methods are often trained and validated on benchmark datasets comprising of single annotated images [223] and videos [222, 224]. The vast amount of objects in these datasets are all highly different but have in common that they represent typical objects seen in normal life. In additive manufacturing, and DED melt-pool segmentation specifically, the molten material is segmented from the solidified material. These object domains are highly different, with differences in segmentation boundary ambiguity, object geometry, object orientation, and occlusions as studied in [145].

This work studies the application of SVOS as well as IVOS for DED melt-pool segmentation. Although the current state-of-the-art and fastest VOS implementation is able to reach suitable inference latency in the SVOS setting, performance decreases whenever more memory is utilized. The complexity of their VOS architecture is sufficiently capable for a variety of objects, while VOS for melt-pool segmentation likely does not require it, due to the more constraint operational environment. We therefore introduce a novel VOS architecture, called AMVOS, in Section 5.3.3, specifically tailored for AM. Our solution is focused on reducing latency and variable re-use to reduce model memory storage occupation.

5.3.3 AMVOS: VOS specifically for AM

Our VOS solution, which is specifically tailored for the inference requirements necessary for real-time melt-pool segmentation of DED footage, is generally applicable beyond AM and DED to any object segmentation task, and image input dimension, for any number of objects. The following paragraphs provide insights in the most significant changes AMVOS incorporates compared to existing VOS methods [220, 221]. A complete description of the AMVOS architecture can be found in the Supplementary material 5.3.6, Figure D.1 and Figure D.2.

We use a single (instead of two) CNN encoder for image and object ground-truth/prediction encoding to save memory occupation, and reduce the total number of parameters, reducing the likelihood of over-fitting. Since object ground truth and prediction mask have shape $[H, W, 1]$ with the first two dimensions referring to the height and width, we repeat the prediction mask along the channel dimension to have similar shape as the input images $[H, W, 3]$. Instead of the older ResNet-50 and MobileNetV2 CNN encoders used by authors in their recent work [220, 221, 219], we employ the more modern MobileNetV3 architecture [225], pre-trained on the Imagenet dataset since it is faster on CPU and GPU platforms. Only part of MobileNetV3 is used, where two intermediate features are selected as encoder outputs, with dimensions $[H/16, W/16, 112]$ and $[H/4, W/4, 40]$. The former being used by the cross attention module and the latter used as skip connection to the decoder.

The decoders used in prior literature [110, 218, 220, 221, 219], combine the features from the cross attention module with 2 skip connection outputs from the encoders. In 3 up-sampling stages, the features are combined using three 1×1 convolutional blocks and five 3×3 convolutional residual blocks with each residual block consisting out of 2 reLU activation functions and two 3×3 convolutional blocks. Our solution greatly simplifies this design by using a single skip connection, 2 up-sampling stages, and 2 residual blocks, only with each block consisting of two 3×3 convolutional blocks and a single reLU.

The cross attention module is inspired by the transformer architecture described by Vaswami et al. [226]. The lowest spatial dimension features from the image and mask encoders are turned into keys and values respectively using two 3×3 convolutional blocks. The keys from the current frame are matched with keys from images stored in memory using scaled dot-product attention and turned into probabilities using the *softmax* function. The probability matrix is multiplied with the value features, that combines feature along the spatial and temporal dimension. AMVOS, simplifies this design by taking away the key and value encoding convolutional blocks and using the lowest dimensional encoder output $[H/16, W/16, 112]$ directly as key and value at the same time. Furthermore, using the

softmax probability map, we combine not only mask encodings, but image encodings as well. This added complexity is deemed necessary for object specific model training, where object information is stored in the parameters.

5.3.4 Operation modes

VOS training and validation are typically conducted with a single initial image-mask pair, similar to the SVOS setting. However, AMVOS and VOS architectures, in general, offer flexibility for various modes of operation, primarily influenced by the memory management policies. In the general semi-supervised setting (SVOS), initial ground truth $[(I_1, M_1), \dots, (I_N, M_N)]$ and previous prediction $[(I_{N+1}, P_{N+1}), \dots, (I_K, P_K)]$ information is ideally available for the next prediction P_{K+1} . Since memory occupation would grow infinitely large, different policies have been introduced to limit memory. The most popular policy is to store encoded information from the first initial hint and last previous prediction $[(I_1, M_1), (I_K, M_K)]$ for current prediction P_{K+1} given image I_{K+1} [220, 221]. AMVOS utilizes an first-in-first-out (FIFO) policy for the prediction memory to limit memory occupation. The size q_P is indicating the maximum size of the FIFO queue, while the size of initially provided image-mask pairs is referred to as q_A from here on. These runtime parameters can be changed and experimented with during operation. Implementation details are provided in the Supplementary material Figure D.2.

In this work we investigate two different operating modes for the AMVOS architecture. AMVOS-A, see Figure 5.1, refers the object agnostic mode, which is equivalent to the traditional SVOS setting, where ground-truth data $q_A > 0$ is provided initially. Training and inference in this mode forces the AMVOS architecture to be object agnostic, as the segmentation output is dependent on the initially provided information. During AMVOS-A training, the network is prevented to possess object specific knowledge. Besides object agnostic operation, it is possible to train and use the AMVOS architecture in an object specific manner which we call the AMVOS-S operating mode, as presented in Figure 5.2. AMVOS-S, in contrast to AMVOS-A, is operated without initial ground-truth information $q_A = 0$ and therefore requires the network parameters to learn object specific knowledge as the first prediction has no previous information to attend to.

IVOS is traditionally performed using scribbles instead of complete ground truth masks [222]. In this study we however perform full ground-truth mask IVOS, called AMVOS-I from here on. AMVOS-I is similar to training-free active learning methods [227], where iteratively, an image sequence can be annotated and potentially used to train other ML architectures. A visual representation of AMVOS-I is provided in Figure 5.3. AMVOS-I

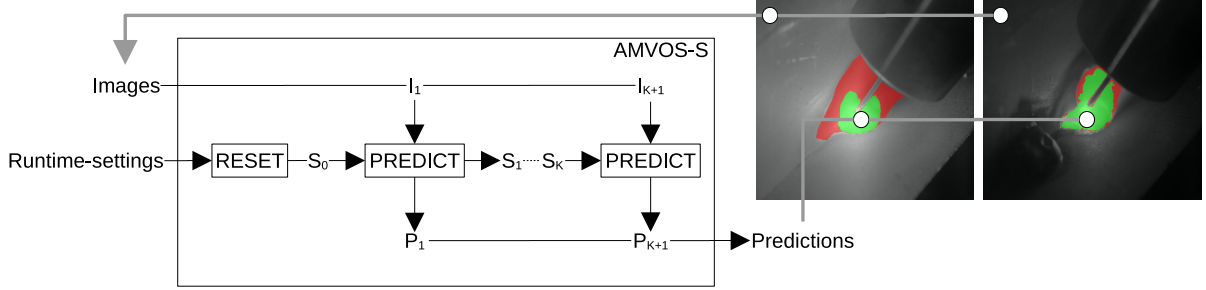


Figure 5.2: AMVOS-S; Object specific video object segmentation without initial hints.

uses AMVOS-A as a recommendation system. Initially, a set of ground truth information, and a sequence of images is provided to AMVOS-A. The output sequence of predictions is thereafter used by a selection algorithm which selects a subset of the predictions. This subset is thereafter annotated by an oracle, typically a human, and added to the already existing set of ground truth information. The selection algorithm used in this study is based on the information entropy in the predictions, related to prediction uncertainty and is specified as:

$$uncertainty(P) = - \sum_{i \in O \cdot H \cdot W} P(i) \cdot \log(P(i)) \quad (5.1)$$

with M the ground truth masks, P the prediction, O the number of objects, and H , W the height and width respectively. Index i refers to a specific spatial and object location in the prediction. A subset can then be formed by selecting the q_I predictions with the highest uncertainty.

5.3.5 Datasets

AMVOS operation in an object agnostic manner (AMVOS-A), enables transfer learning and exploiting different datasets for training purposes, while object specific operation (AMVOS-S) is tied to single object type datasets. Object agnostic operation can therefore have a significant advantage over object specific operation. To explore the effect of transfer learning, AMVOS operation modes, and memory policies, we utilize several datasets. An overview of these datasets is provided in Table 5.3.5.

The first, and most important dataset, used for training and validation in the object agnostic and specific setting, is called the XIRIS dataset. From a set of 49 different DED machines, from now on referred to as "scenes", a collection of 178 in-process videos

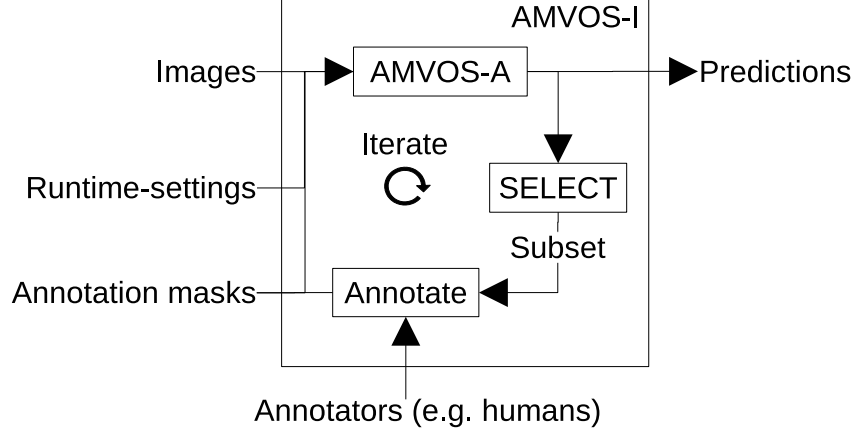


Figure 5.3: AMVOS-I; Interactive video object segmentation (IVOS) using full ground-truth annotations.

Table 5.1: AMVOS training and validation datasets

Dataset	Videos	Images	Objects
XIRIS(train)	163	2403	163
XIRIS(validation)	15	763	15
COCO (2017) [223]	-	99787	592985
YOUTUBE-VOS (2018) [224]	3468	94496	6025
DAVIS (2017) [222]	90	6208	206

is acquired, all using different, but unknown, process parameters, camera settings, and materials, as these videos were acquired from external sources without further information. Due to time-constraints a subset of 3166 images from all scenes is annotated using the AWUS active learning method as introduced by van Houtum et al.[145]. Since AWUS leads to the annotation of information that is rich and in general difficult data instances, some videos have little annotated images, while others have many. AMVOS validation ideally requires long image sequences to observe the impact of memory policies, while training does not as VOS architectures are able to train on very short sequences [110]. Furthermore, proper out-of-domain validation requires training and validation on different scenes. From the 49 scenes, 15 are selected for the validation set. From these scenes, the video with the most annotations is selected, resulting in a validation dataset of 15 videos, with 16 to 121

annotated melt-pool images. The rest of the videos, totalling 2403 images, are selected for training. The annotation protocol used to create melt-pool boundary polygons is provided in Figure A.1 of Supplementary material A.

Object agnostic operation enables transfer learning through AMVOS training on any segmentation dataset from static images, to annotated videos. The COCO dataset [223], is a common dataset used in VOS training. Small objects are typically removed from the dataset, leading to a total of 95,787 images in our implementation, with a total of 592,985 different objects. COCO therefore adds significantly to the object diversity AMVOS-A experiences compared to the XIRIS dataset, when trained upon. Synthetic image sequences are created by repeating the static images. Data augmentation is further discussed in Section 5.3.6. Additionally, we train on the DAVIS 2017 [222] and YOUTUBE-VOS 2018 [224] video sequence datasets. These datasets have multiple annotated objects per video. In total DAVIS 2017 comprises of 90 videos with 25 to 104 annotated images, and 6,208 images in total. YOUTUBE-VOS is much larger with 3,468 videos, 4 to 36 images per video, and 94,496 images in total. These number differ slightly from the original size, as small objects and very short (< 5) sequences are removed.

5.3.6 Training experiments

Object specific AMVOS-S is trained solely on the XIRIS dataset, as the other datasets cannot be used for transfer learning. Object agnostic AMVOS-A, is trained on XIRIS alone, and COCO, DAVIS, YOUTUBE-VOS, followed by XIRIS to study the impact of transfer learning. Training solely on the XIRIS dataset allows to directly compare the operating modes, while using the same AMVOS architecture. The MobileNetV3 encoder is initialized with Imagenet [190] pre-trained weights before any training.

To train object specific and agnostic AMVOS, we create image sequences of length 5 just as previously proposed by Miles et al. [220]. From the COCO dataset a single image is repeated, while for the XIRIS, DAVIS and YOUTUBE-VOS datasets a window of 5 sequential frames in the videos are selected at uniform random. A batch size of 8 is used, combining 8 sequences of 5 image-ground-truth mask pairs, and 1 object is selected at uniform random that is present in the first frame as the initial hint.

The image resolution of all datasets differs and are initially resized to $H = W = 480$, with H the height and W the width, using bi-linear and nearest neighbor down-sampling for the image and masks respectively. The RGB images are normalized between 0 and 1 through division by 255 and different data augmentation protocols are applied to the images similar to previous works [220, 221] to mitigate over-fitting. Random horizontal

and vertical flipping and random 90 degree rotation is applied the same to each image in a sequence. For each image separately, random 30-100 % cropping, random contrast 0-50 % and random brightness 0-50% adjustments are applied. Cropping selects a random patch in the image and resales it back to the $H = W = 480$ resolution.

To train AMVOS, the sequential segmentation steps are unfolded. In case of object specific AMVOS-S, all 8×5 images in the augmented batches are segmented sequentially without any initial provided image-ground truth memory. Agnostic AMVOS-A uses the first image-mask pair in each sequence as the initial hint and segments the other images in the batch. Both settings store all previous image-prediction mask pairs in memory for the next prediction.

A categorical cross-entropy loss is used to minimize the difference between the ground truth masks M and the predicted object probability masks P generated by AMVOS. The total loss is a sum of all cross entropy losses over the batch of prediction sequences combined. The loss of a single prediction is defined as:

$$loss(M, P) = - \sum_{i \in O \cdot H \cdot W} M(i) \cdot \log(P(i)) \quad (5.2)$$

5.3.7 Validation experiments

AMVOS performance validation is performed for the AMVOS-A, AMVOS-S and AMVOS-I operating modes on the XIRIS validation dataset. Additionally, run-time analysis is investigated. Semi-supervised (SVOS) performance is evaluated for the AMVOS-A and AMVOS-S architectures trained on different datasets as discussed in Section 5.3.6.

The SVOS validation experiments are driven by different memory management policies for the prediction memory for both AMVOS-A and AMVOS-S. The effect of different FIFO prediction memory queue sizes q_P is investigated to determine an optimal latency-performance trade-off for DED melt-pool segmentation. Performance is measured by calculating the average intersection-over-union (IOU) metric for each object in a sequence, over all objects, equivalent to the work by Perazzi et al. [222] with M and P the ground-truth masks and predictions respectively of object o in image sequence V_o :

$$metric = \frac{1}{|O|} \sum_{o \in O} \frac{1}{|V_o|} \sum_{M, P \in V_o} IOU(M, P) \quad (5.3)$$

To study the effect full-feedback IVOS on the performance on the XIRIS validation dataset, several AMVOS-I iterations are performed. From the SVOS experiments, several

q_P sizes, and their effect on performance, are investigated throughout the AMVOS-I experiments.

5.3.8 Implementation details

Training is conducted using a single Nvidia RTX 3090 GPU. XIRIS dataset training takes about 2 hours for 60,000 batch iterations, while pre-training on the other datasets (DAVIS, YOUTUBE, COCO) takes about 5 hours for 150,000 batch iterations. For numerical stability, we train using mixed precision, leaving variables 32-bit, while performing computation in 16-bit to speed up training. A static loss scale of 1000 is used to prevent gradient over and underflow. To further decrease training time, and memory occupation during training, XLA compilation is used. We use the RMSprop optimizer, with a learning rate of $1e-4$, and use gradient clipping with a clip value of 1 to be able to run a higher learning rate. Inference is conducted in 32-bit without XLA.

Our implementation of AMVOS is created from scratch in Tensorflow [228], while all other works in VOS are derived from a single implementation in Pytorch [229]. We consider this a significant practical contribution to the field of VOS on its own as it will potentially enable research and development of new VOS solutions in the Tensorflow framework.

5.4 Results

This section starts with a running time analysis of the AMVOS architecture described in Section 5.3.3, followed by semi-supervised performance evaluations for the AMVOS-A and AMVOS-S operating modes described in Section 5.3.4. At last, performance is investigated for interactive video object segmentation using the AMVOS-I training-free active learning method. The results indicate that AMVOS-A offers superior performance compared to AMVOS-S, and that interactive AMVOS-I leads to an additional performance increase.

5.4.1 Running time analysis

The architectural changes made in AMVOS compared to existing VOS architectures [220, 221] as discussed in Section 5.3.3, are aimed at reducing inference latency for the requirements of AM and DED as discussed in Section 5.3.2. The influence of the size of memory features maintained during prediction is investigated through multiple

experimental runs with different total memory FIFO queue sizes as the relative queue size between initial q_A and prediction memory q_P is not affecting performance. Whether a prediction memory is used ($q_P > 0$) or not ($q_P = 0$), does however impact performance. Without prediction memory updating, prediction encoding and memory updating is prevented, resulting in less computations as shown in Figure D.1 and Figure D.2 of the Supplementary material.

Inference latency results are presented in Figure 5.4, with ($q_P \neq 0$) and without ($q_P = 0$) prediction memory for different sizes of the total memory. For evaluation, an image resolution of $H = 480, W = 854$ is used, which is similar to other literature studies [220, 110, 221]. Without memory ($q_A = q_P = 0$), AMVOS operates as a simple encoder-decoder image segmentation network without cross-attention, only applicable for object specific operation, or AMVOS-S. The lowest latency is achieved in this setting of 323 frames per second (FPS). With prediction memory ($q_P \neq 0$), AMVOS latency increases drastically relative to operation without ($q_P = 0$). Total memory size of 1 with $q_P = 0$ and $q_A = 1$ is only possible in the AMVOS-A setting achieving 162 FPS, while $q_P = 1$ and $q_A = 0$ with 261 FPS refers to the AMVOS-S mode only. Larger amounts of memory ($q_A + q_P > 1$) lead to an increase in latency.

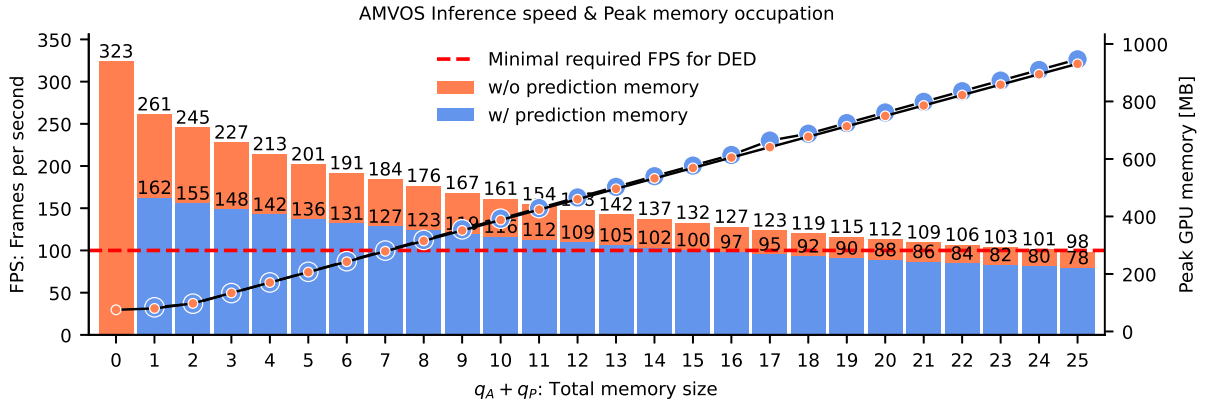


Figure 5.4: Speed and memory occupation of AMVOS between inference without and with prediction memory using an image resolution of $[H = 480, W = 854]$ on 32-bit RTX 3090 GPU.

Considering the requirements for VOS operation in AM, and DED specifically, as discussed in 5.3.2, a total memory of size $q_A + q_P < 16$ and $q_A + q_P < 25$ are required for operation with and without prediction memory respectively to achieve 100+ FPS. Therefore providing room for experimentation of different values of q_A and q_P in the

SVOS and IVOS settings respectively. Memory occupation as shown in Figure 5.4 shows that the additional computation required when $q_P \neq 0$ does not significantly affect peak GPU memory occupation, providing only a small overhead. A large memory size ($q_A + q_P = 25$) does however require significant more memory (≈ 1 GB) compared to a small memory size ($q_A + q_P = 2$) with ≈ 40 MB.

5.4.2 Semi-supervised performance

Semi-supervised VOS operation (SVOS) of object agnostic AMVOS-A, and object specific AMVOS-S, as discussed in Section 5.3.4 are compared for different prediction memory queue sizes q_P whenever trained on different training protocols as discussed in Section 5.3.6. Semi-supervision is provided to AMVOS-A by providing the first frame and object mask $q_A = 1$ while employing different prediction memory queue sizes q_P . AMVOS-S is evaluated without initial hint, as it does not require it. SVOS performance is shown in Figure 5.5 for the 4 different trained models created using the AMVOS architecture. AMVOS-A(XIRIS) and AMVOS-S(XIRIS) are trained on the XIRIS dataset without pre-training on other object datasets (DAVIS, YOUTUBE-VOS, COCO), from here on referred to as OTHER. AMVOS-A(OTHER) is pre-trained only and AMVOS-A(OTHER $\rightarrow$ XIRIS) is AMVOS-A(OTHER) fine-tuned on the XIRIS dataset.

The overall results show that a relatively high IOU of 80+ % is achieved for all AMVOS models trained on the XIRIS dataset, while AMVOS-A(OTHER) shows a significant lower performance indicating the importance of observing melt-pool related objects during training. Another important observation is that without prediction memory ($q_P = 0$), AMVOS-A and AMVOS-S under-perform significantly compared to using a last prediction only memory ($q_P = 1$) as the models show a drop in IOU ranging from 1.0 to 4.1%. Larger prediction memory sizes ($q_P > 1$) show improved performance compared to last only ($q_P = 1$) for all models with gains ranging from 0.3 to 0.6% IOU. The gains are however relatively small, indicating the effectiveness of last only memory ($q_P = 1$), especially when inference latency is taken in consideration as discussed in Section 5.4.1.

Pre-training AMVOS-A on OTHER results in 70.3 to 73.7 % IOU indicating that knowledge gained during training on other objects transfers to DED melt-pool data. Object specific training in AMVOS-S leads to 80+% IOU for $q_P > 0$ while not improving significantly for larger prediction memory sizes. Object agnostic AMVOS-A without pre-training shows superior performance to AMVOS-S whenever trained solely on the XIRIS train dataset by achieving 81.8 to 82.3% IOU indicating the importance of initial

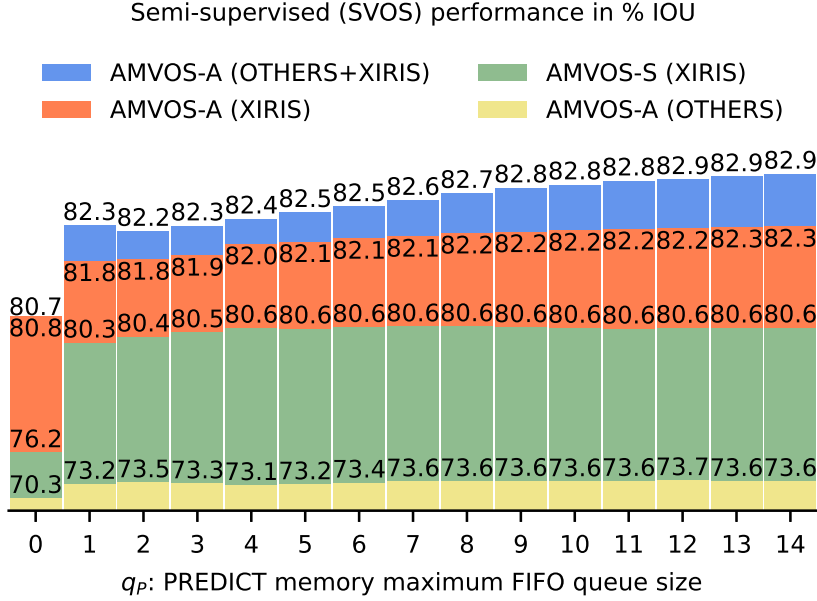


Figure 5.5: AMVOS; semi-supervised performance analysis

ground truth information as AMVOS-S is operated without initial memory ($q_A = 0$). The best performance is achieved by fine-tuning AMVOS-A(OTHERS) on the XIRIS train dataset as AMVOS-A reaches 82.2 to 82.9 % IOU.

5.4.3 Interactive performance

Interactive VOS operation of AMVOS-I is investigated for object agnostic AMVOS-A, which is pre-trained on OTHER, and fine-tuned on the XIRIS dataset. The other trained models as discussed in Section 5.4.2 are omitted as they under-perform in the AMVOS-I setting. Experiments are conducted without initial memory, so $q_A = 0$. Iteratively, the most uncertain single prediction is selected, the mask prediction is replaced with the original ground-truth annotation, and added to the existing memory leading to a memory of size $q_A = 1$ after 1 iteration. This process is repeated for several iterations. As shown in Figure 5.5, prediction memory size q_P has a large impact on performance, therefore several experiments are conducted using different values of q_P .

AMVOS-I results are presented in Figure 5.6. Inference using a large prediction memory of $q_P = 15$ shows the best performance, which is expected, as the most historical

information is maintained during operation. An interesting observation is the difference in performance between choosing the first frame and ground truth mask as initial hint, which shows 82.3 % IOU in Figure 5.5, versus the initial choice AMVOS-I made with 83.2 % IOU. The choice of initial information provided to AMVOS-A is therefore affecting performance and should be chosen carefully. Each AMVOS-I iteration is leading to better AMVOS-A performance leading to 83.4 % IOU at $q_P = 15, q_A = 14$. A total memory of $q_A + q_P = 29$ is however too slow to operate for DED processes as shown in Figure 5.4. Iteratively adding ground truth information using AMVOS-I generally leads to increased performance each iteration, but only for larger prediction memory sizes $q_P > 7$. Last only prediction memory $q_P = 1$ shows erratic behaviour as AMVOS-I does not contribute to performance after each iteration. Without prediction memory $q_P = 0$, AMVOS-I does not add to the performance at all. AMVOS-I is therefore useful under the constraint that a larger prediction memory is maintained.

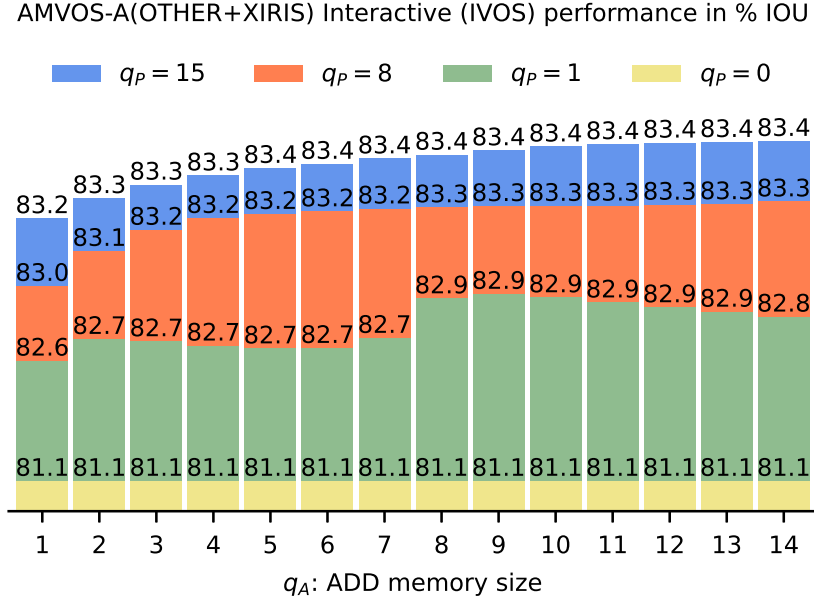


Figure 5.6: AMVOS-I; interactive performance analysis

5.5 Discussion

This section provides the explanation and interpretation of the results presented in Section 5.4. First the inference runtime is compared to prior studies, followed by a discussion on the implications and limitations of the performance results.

5.5.1 Running time comparison

AMVOS shows superior inference speed performance whenever compared to existing state-of-the-art VOS methods as shown in Table 5.5.1. The architecture and platform used by Skartados et al.[221] is most comparable to AMVOS. AMVOS is however almost $1.5x$ faster and uses less than half the number of parameters indicating the effectiveness of the architectural changes introduced in Section 5.3.3.

Table 5.2: Parameter count and inference speed comparison with state-of-the-art VOS (Memory size = 2)

VOS architecture	Parameters (M)	FPS (GPU)
STM (2019) [110]	38.9	6.3 (1080ti)
STCN (2021) [218]	54.4	26.9 (2080ti)
SwiftNet (2021) [219]	-	70 (P100)
MobileVOS (2023) [220]	2.5	99 (A10)
TrickVOS (2023) [221]	1.9	102 (RTX 3090)
AMVOS (ours)	0.9	155 (RTX 3090)

AMVOS is able to operate at sufficient speed for DED (100+ FPS) for total memory sizes up to $q_P + q_A < 25$ without prediction memory $q_P = 0$, and $q_A + q_P < 16$ with prediction memory $q_P \neq 0$. Even-though $q_P = 0$ provides a significant inference speed advantage over $q_P > 0$, it leads to significant worse performance as shown in Figures 5.5 and 5.6. AMVOS provides room to experiment with different memory policies, which is something that prior studies [220, 221] do not provide as they are only able to operate using a total memory size of $q_A + q_P < 3$.

5.5.2 Memory policy performance influence

Maintaining a larger set of historical information generally leads to better performance and increased latency, as discussed in Sections 5.4.2 and 5.4.3. The memory policy determined

by the hint (q_A) and prediction (q_P) memory queue sizes has a significant effect on AMVOS operation in terms of robustness and performance. Potentially the biggest insight is that adding initial hint memory q_A using AMVOS-I only works well for larger prediction memory sizes q_P . A last only prediction memory policy $q_P = 1$ leads to erratic prediction behaviour while larger sizes lead to more stable results. A potential cause could be that maintaining a larger temporal history could provide AMVOS with information related to velocity and acceleration of objects in the field-of-view of the images provided. Without prediction memory, no temporal information is available, leaving AMVOS-I operation useless. A general advice is therefore to use a prediction memory as large as possible without violating the DED inference latency requirement of 100 FPS.

The performance of AMVOS on the XIRIS validation dataset seems to stagnate at around 83 % IOU of which prediction and annotation ambiguity could be the cause as discussed in 5.3.2. Two temporally close images with annotations and AMVOS predictions, which showed the biggest discrepancy between annotation and prediction is shown in Figure 5.7. It is unlikely that the melt-pool changes size as drastically as depicted in the figure according to the annotations. Furthermore, most errors in the left picture belong to pixels predicted as being melt-pool which seems more plausible than the annotation. The exact opposite is visible in the right picture, where most errors belong to pixels not being predicted as melt-pool. The former and latter are generally considered false-negative and false-positive cases. Whether the annotation or predictions are correct is uncertain with stagnating AMVOS performance as a plausible result.

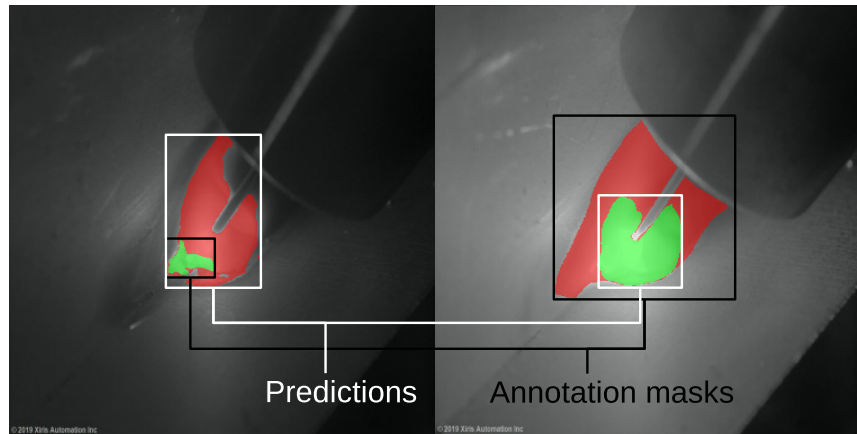


Figure 5.7: Annotation and prediction disagreement. Annotation and prediction agreement is shown in green, disagreement in red, indicating errors. Boxes are determined manually

5.6 Conclusions

This work presents a novel training-free adaptive image segmentation model called AMVOS that can segment any object without requiring re-training. AMVOS is versatile, as it can be trained to be object specific or agnostic. Furthermore, it can be used as a annotation recommendation system, much like active learning, but without the need to re-train. The architecture is developed with real-time deployment on additive manufacturing processes, and specifically directed energy deposition in mind, as it is capable of reaching 83.4 % intersection-over-union at 100+ frames-per-second for larger external memory sizes. The speed of AMVOS is unprecedented in the field of video object segmentation as it is $1.5x$ faster than the competition. Our work can help the additive manufacturing community as the need for large and diverse annotated datasets is minimized as only a few (less than 5) annotations are typically needed for AMVOS to adapt to newly unseen scenery without training. Furthermore, AMVOS can be trained on any image or video sequence dataset, and any object, therefore greatly expanding the available data for training with better generalization and performance as a result.

Chapter 6

Conclusions and future work

6.1 Conclusions

This thesis proposes several advancements in the application of ML tools within AM, and specifically DED. The focus was to enhance both the generalizability and scalability of ML models, ensuring their effectiveness across a wide range of industrial applications and operating conditions. This research was centered around three main objectives: reducing the need for extensive annotated data, developing and evaluating ML models on their cross-domain generalization capabilities, and creating efficient and adaptable solutions for real-time deployment.

In Chapter 3, AWUS is introduced, which is a novel active learning method that balances random exploration with uncertainty based exploitation through model change. AWUS significantly reduces the amount of annotated data required for training ML models, with reductions ranging from 20% to 70% across 36 datasets from various domains, including DED. The ability to leverage random exploration is particularly useful in scenarios where defining image similarity is complex or difficult, such as in-situ process quality prediction for DED. Furthermore, AWUS was used to create the diverse dataset used in Chapter 4. This dataset, consisting of 9,806 images from 285 videos of 49 different DED machines, was essential for validating ML models across different domain shifts.

Chapter 4 presented a novel methodology for validating ML-based DED process quality classification models by simulating various input-distribution domain shifts, likely to be experienced in real-world DED operation. The dataset created with AWUS enabled a comprehensive analysis of model performance in diverse operating environments. Our

findings indicate that supervised vision-based ML models perform relatively well with minor domain shifts, such as small changes in process parameters or tool-path variations. Performance significantly degrades with hardware changes, averaging a -20.9% reduction in classification accuracy. This finding highlighted the need for models capable of adapting to hardware shifts and directly influenced the development of AMVOS. This chapter also reviewed 83 convolutional neural network (CNN) architectures for DED quality prediction, providing valuable insights into model selection based on performance and inference latency.

In Chapter 5, we introduced AMVOS, a training-free adaptive image segmentation model designed for real-time deployment in AM and DED. AMVOS addresses the challenge of performance degradation due to hardware domain shifts by providing a solution that does not require re-training. It achieved an intersection-over-union score of 83.4% at over 100 frames per second, making it 1.5 times faster than existing solutions in video object segmentation. The model’s ability to adapt to new environments with fewer than five annotations significantly enhances its generalizability and deployment efficiency. The development of AMVOS was directly motivated by the need for robust solutions to handle significant domain shifts, particularly those identified in Chapter 4.

6.2 Future work

There are several ways future work can extend this dissertation. One promising direction is the exploration of integrating AWUS and AMVOS into other AM processes beyond DED. Investigating how these methods can be adapted to different additive manufacturing technologies, such as powder bed fusion or binder jetting, could extend the applicability of the contributions to AM.

Additionally, there is potential for combining AWUS and AMVOS with emerging technologies, such as advanced sensor systems or real-time process monitoring tools in industrial environments. Exploring these combinations could lead to innovative solutions that better handle the complexities of modern manufacturing environments.

Improving AWUS and AMVOS could also be valuable. For AWUS, future work could focus on optimizing and validating the balancing between exploration and exploitation to further reduce annotation requirements. For AMVOS, enhancing its capabilities to manage more complex segmentation tasks, such as increased occlusions or object ambiguity, could be helpful in real-world operating scenarios where sub-optimal process parameters are used.

Finally, conducting real-world validation studies to test the developed ML models in

practical industrial settings beyond AM and DED would provide valuable insights into their effectiveness and robustness. These studies could help to refine the models and ensure their applicability in real-world operating scenarios.

In conclusion, this work provides a foundation for data-efficient machine learning solutions in additive manufacturing that could be build upon. By addressing key challenges related to generalizability and scalability, the research has made contributions to the field of AM, and DED specifically. The integration of these findings into future research can continue to drive innovation and progress in ML applications for additive manufacturing.

References

- [1] Saquib Rouf, Abrar Malik, Navdeep Singh, Ankush Raina, Nida Naveed, Md Irfanul Haque Siddiqui, and Mir Irfan Ul Haq. Additive manufacturing technologies: Industrial and medical applications. *Sustainable Operations and Computers*, 3:258–274, 2022.
- [2] Ana Vafadar, Ferdinando Guzzomi, Alexander Rassau, and Kevin Hayward. Advances in metal additive manufacturing: a review of common processes, industrial applications, and current challenges. *Applied Sciences*, 11(3):1213, 2021.
- [3] Mahesh Mani, Shaw Feng, Lane Brandon, Alkan Donmez, Shawn Moylan, and Ronnie Fesperman. *Measurement science needs for real-time control of additive manufacturing powder-bed fusion processes*. CRC Press, 2017.
- [4] Sayyeda Saadia Razvi, Shaw Feng, Anantha Narayanan, Yung-Tsun Tina Lee, and Paul Witherell. A review of machine learning applications in additive manufacturing. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 59179, page V001T02A040. American Society of Mechanical Engineers, 2019.
- [5] Guo Dong Goh, Swee Leong Sing, and Wai Yee Yeong. A review on machine learning in 3d printing: Applications, potential, and challenges. *Artificial Intelligence Review*, 54(1):63–94, 2021.
- [6] Chengcheng Wang, XP Tan, Shu Beng Tor, and CS Lim. Machine learning in additive manufacturing: State-of-the-art and perspectives. *Additive Manufacturing*, page 101538, 2020.
- [7] Alejandro Germán Frank, Lucas Santos Dalenogare, and Néstor Fabián Ayala. Industry 4.0 technologies: Implementation patterns in manufacturing companies. *International Journal of Production Economics*, 210:15–26, 2019.

- [8] Jinjiang Wang, Yulin Ma, Laibin Zhang, Robert X Gao, and Dazhong Wu. Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48:144–156, 2018.
- [9] ISO/ASTM 52900: 2015 (E). Standard terminology for additive manufacturing—general principles—terminology. -, 2016.
- [10] Mark Armstrong, Hamid Mehrabi, and Nida Naveed. An overview of modern metal additive manufacturing technology. *Journal of Manufacturing Processes*, 84:1001–1029, 2022.
- [11] Andrey Zakirov, Sergei Belousov, Maria Bogdanova, Boris Korneev, Andrey Stepanov, Anastasia Perepelkina, Vadim Levchenko, Andrey Meshkov, and Boris Potapkin. Predictive modeling of laser and electron beam powder bed fusion additive manufacturing of metals at the mesoscale. *Additive Manufacturing*, 35:101236, 2020.
- [12] Michal Krzyzanowski and Dmytro Svyetlichnyy. A multiphysics simulation approach to selective laser melting modelling based on cellular automata and lattice boltzmann methods. *Computational Particle Mechanics*, pages 1–17, 2021.
- [13] Saad A Khairallah, Andrew T Anderson, Alexander Rubenchik, and Wayne E King. Laser powder-bed fusion additive manufacturing: Physics of complex melt flow and formation mechanisms of pores, spatter, and denudation zones. *Acta Materialia*, 108:36–45, 2016.
- [14] WA Ayoola, WJ Suder, and Stewart W Williams. Parameters controlling weld bead profile in conduction laser welding. *Journal of Materials Processing Technology*, 249:522–530, 2017.
- [15] Tarasankar DebRoy, HL Wei, JS Zuback, T Mukherjee, JW Elmer, JO Milewski, Allison Michelle Beese, A Wilson-Heid, A De, and W Zhang. Additive manufacturing of metallic components—process, structure and properties. *Progress in Materials Science*, 92:112–224, 2018.
- [16] Byron Blakey-Milner, Paul Gradl, Glen Snedden, Michael Brooks, Jean Pitot, Elena Lopez, Martin Leary, Filippo Berto, and Anton Du Plessis. Metal additive manufacturing in aerospace: A review. *Materials & Design*, 209:110008, 2021.
- [17] Rachel Lacroix, Ralf W Seifert, and Anna Timonina-Farkas. Benefiting from additive manufacturing for mass customization across the product life cycle. *Operations Research Perspectives*, 8:100201, 2021.

- [18] Andrea Zocca, Janka Wilbig, Anja Waske, Jens Günster, Martinus Putra Widjaja, Christian Neumann, Mélanie Clozel, Andreas Meyer, Jifeng Ding, Zuoxin Zhou, et al. Challenges in the technology development for additive manufacturing in space. *Chinese Journal of Mechanical Engineering: Additive Manufacturing Frontiers*, 1(1):100018, 2022.
- [19] Aprilia Aprilia, Naien Wu, and Wei Zhou. Repair and restoration of engineering components by laser directed energy deposition. *Materials Today: Proceedings*, 70:206–211, 2022.
- [20] Brian J Ree. Critical review and perspectives on recent progresses in 3d printing processes, materials, and applications. *Polymer*, page 127384, 2024.
- [21] Jungeon Lee, Hyung Jun Park, Seunghak Chai, Gyu Ri Kim, Hwanwoong Yong, Suk Joo Bae, and Daeil Kwon. Review on quality control methods in metal additive manufacturing. *Applied Sciences*, 11(4):1966, 2021.
- [22] Adrita Dass and Atieh Moridi. State of the art in directed energy deposition: From additive manufacturing to materials design. *Coatings*, 9(7):418, 2019.
- [23] Hossein Taheri, MRBM Shoaib, Lucas W Koester, Timothy A Bigelow, Peter C Collins, and Leonard J Bond. Powder based additive manufacturing-a review of types of defects, generation mechanisms, detection, property evaluation and metrology. *Int. J. Addit. Subtractive Mater. Manuf*, 1(2):172–209, 2017.
- [24] Zhong Yang Chua, Il Hyuk Ahn, and Seung Ki Moon. Process monitoring and inspection systems in metal additive manufacturing: Status and applications. *International Journal of Precision Engineering and Manufacturing-Green Technology*, 4(2):235–245, 2017.
- [25] William E Frazier. Metal additive manufacturing: a review. *Journal of Materials Engineering and Performance*, 23(6):1917–1928, 2014.
- [26] Subrata Mondal, CP Paul, LM Kukreja, Asish Bandyopadhyay, and Pradip Kumar Pal. Application of taguchi-based gray relational analysis for evaluating the optimal laser cladding parameters for aisi1040 steel plane surface. *The International Journal of Advanced Manufacturing Technology*, 66(1-4):91–96, 2013.
- [27] Dong-Gyu Ahn. Directed energy deposition (ded) process: State of the art. *International Journal of Precision Engineering and Manufacturing-Green Technology*, pages 1–40, 2021.

- [28] Ross Cunningham, Sneha P Narra, Colt Montgomery, Jack Beuth, and AD Rollett. Synchrotron-based x-ray microtomography characterization of the effect of processing variables on porosity formation in laser power-bed additive manufacturing of ti-6al-4v. *Jom*, 69(3):479–484, 2017.
- [29] DB Hann, J Iammi, and J Folkes. A simple methodology for predicting laser-weld properties from material and laser parameters. *Journal of Physics D: Applied Physics*, 44(44):445401, 2011.
- [30] Shyam Barua, Frank Liou, Joseph Newkirk, and Todd Sparks. Vision-based defect detection in laser metal deposition process. *Rapid Prototyping Journal*, 2014.
- [31] CR Heiple, JR Roper, RT Stagner, and RJ Aden. Surface active element effects on the shape of gta, laser and electron beam welds. *Weld. J.*, 62(3):72, 1983.
- [32] Amanda S Wu, Donald W Brown, Mukul Kumar, Gilbert F Gallegos, and Wayne E King. An experimental investigation into additive manufacturing-induced residual stresses in 316l stainless steel. *Metallurgical and Materials Transactions A*, 45(13):6260–6270, 2014.
- [33] Alexander Kaplan. Metallurgy and imperfections of welding and hardening. In *The Theory of Laser Materials Processing*, pages 241–261. Springer, 2017.
- [34] J Gockel, J Fox, J Beuth, and R Hafley. Integrated melt pool and microstructure control for ti-6al-4v thin wall additive manufacturing. *Materials Science and Technology*, 31(8):912–916, 2015.
- [35] William J Sames, FA List, Sreekanth Pannala, Ryan R Dehoff, and Sudarsanam Suresh Babu. The metallurgy and processing science of metal additive manufacturing. *International Materials Reviews*, 61(5):315–360, 2016.
- [36] Luke N Carter, Moataz M Attallah, and Roger C Reed. Laser powder bed fabrication of nickel-base superalloys: influence of parameters; characterisation, quantification and mitigation of cracking. *Superalloys*, 2012(6):2826–2834, 2012.
- [37] AFH Kaplan and John Powell. Spatter in laser welding. *Journal of laser Applications*, 23(3):032005, 2011.
- [38] WJ Sames, F Medina, WH Peter, SS Babu, and RR Dehoff. Effect of process control and powder quality on inconel 718 produced using electron beam melting. In *8th Int. Symp. Superalloy 718 Deriv*, pages 409–423, 2014.

- [39] RK Nalla, RO Ritchie, BL Boyce, JP Campbell, and JO Peters. Influence of microstructure on high-cycle fatigue of ti-6al-4v: Bimodal vs. lamellar structures. *Metallurgical and Materials Transactions A*, 33(3):899–918, 2002.
- [40] Allison M Beese and Beth E Carroll. Review of mechanical properties of ti-6al-4v made by laser-based additive manufacturing using powder feedstock. *Jom*, 68(3):724–734, 2016.
- [41] Nima Shamsaei, Aref Yadollahi, Linkan Bian, and Scott M Thompson. An overview of direct laser deposition for additive manufacturing; part ii: Mechanical behavior, process parameter optimization and control. *Additive Manufacturing*, 8:12–35, 2015.
- [42] Jon Iñaki Arrizubieta, Aitzol Lamikiz, Fritz Klocke, Silvia Martínez, Kristian Arntz, and Eneko Ukar. Evaluation of the relevance of melt pool dynamics in laser material deposition process modeling. *International Journal of Heat and Mass Transfer*, 115:80–91, 2017.
- [43] Qian Wang, Jianyi Li, Michael Gouge, Abdalla R Nassar, Edward W Reutzel, et al. Physics-based multivariable modeling and feedback linearization control of melt-pool geometry and temperature in directed energy deposition. *Journal of Manufacturing Science and Engineering*, 139(2), 2017.
- [44] Qianru Wu, Jiping Lu, Changmeng Liu, Hongli Fan, Xuezhi Shi, Jie Fu, and Shuyuan Ma. Effect of molten pool size on microstructure and tensile properties of wire arc additive manufacturing of ti-6al-4v alloy. *Materials*, 10(7):749, 2017.
- [45] Yiming Huang, Shuaishuai Hou, Tao Sun, Lijun Yang, Shanshan Li, and Shanben Chen. An improved model of porosity formation during pulsed gta welding of aluminum alloys. *Materials Science and Engineering: B*, 238:122–129, 2018.
- [46] Sergio Ríos, Paul A Colegrove, Filomeno Martina, and Stewart W Williams. Analytical process model for wire+ arc additive manufacturing. *Additive Manufacturing*, 21:651–657, 2018.
- [47] Yung C Shin, Neil Bailey, Christopher Katinas, and Wenda Tan. Predictive modeling capabilities from incident powder and laser to mechanical properties for laser directed energy deposition. *Computational Mechanics*, 61(5):617–636, 2018.
- [48] N Tamanna, R Crouch, IR Kabir, and S Naher. An analytical model to predict and minimize the residual stress of laser cladding process. *Applied Physics A*, 124(2):1–5, 2018.

- [49] Florian Wirth and Konrad Wegener. A physical modeling and predictive simulation of the laser cladding process. *Additive Manufacturing*, 22:307–319, 2018.
- [50] Yuze Huang, Mir Behrad Khamesee, and Ehsan Toyserkani. A new physics-based model for laser directed energy deposition (powder-fed additive manufacturing): From single-track to multi-track and multi-layer. *Optics & Laser Technology*, 109:584–599, 2019.
- [51] Richard H van Blitterswijk, Mohammad H Farshidianfar, and Amir Khajepour. Model predictive control for real-time microstructure control in laser heat treatment. *Journal of Manufacturing Processes*, 118:116–127, 2024.
- [52] Scott M Thompson, Linkan Bian, Nima Shamsaei, and Aref Yadollahi. An overview of direct laser deposition for additive manufacturing; part i: Transport phenomena, modeling and diagnostics. *Additive Manufacturing*, 8:36–62, 2015.
- [53] Baldev Raj, Tammana Jayakumar, and M Thavasimuthu. *Practical non-destructive testing*. Woodhead Publishing, 2002.
- [54] Sandeep Kumar Dwivedi, Manish Vishwakarma, and Akhilesh Soni. Advances and researches on non destructive testing: A review. *Materials Today: Proceedings*, 5(2):3690–3698, 2018.
- [55] Sarah K Everton, Matthias Hirsch, Petros Stravroulakis, Richard K Leach, and Adam T Clare. Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing. *Materials & Design*, 95:431–445, 2016.
- [56] Visible Light and Gamma Rays. The electromagnetic spectrum. -, 2005.
- [57] Sarah J Wolff, Hao Wu, Niranjana Parab, Cang Zhao, Kornel F Ehmann, Tao Sun, and Jian Cao. In-situ high-speed x-ray imaging of piezo-driven directed energy deposition additive manufacturing. *Scientific reports*, 9(1):1–14, 2019.
- [58] Adam Thompson, Ian Maskery, and Richard K Leach. X-ray computed tomography for additive manufacturing: a review. *Measurement Science and Technology*, 27(7):072001, 2016.
- [59] Lee Aucott, Hongbiao Dong, Wajira Mirihanage, Robert Atwood, Anton Kidess, Shian Gao, Shuwen Wen, John Marsden, Shuo Feng, Mingming Tong, et al. Revealing internal flow behaviour in arc welding and additive manufacturing of metals. *Nature communications*, 9(1):1–7, 2018.

- [60] Leonardo De Chiffre, Simone Carmignato, J-P Kruth, Robert Schmitt, and Albert Weckenmann. Industrial applications of computed tomography. *CIRP annals*, 63(2):655–677, 2014.
- [61] Célia Millon, Arnaud Vanhoye, Anne-Françoise Obaton, and Jean-Daniel Penot. Development of laser ultrasonics inspection for online monitoring of additive manufacturing. *Welding in the World*, 62(3):653–661, 2018.
- [62] Alexia Chabot, N Laroche, E Carcreff, Matthieu Rauch, and J-Y Hascoët. Towards defect monitoring for metallic additive manufacturing components using phased array ultrasonic testing. *Journal of Intelligent Manufacturing*, pages 1–11, 2019.
- [63] Yashar Javadi, Charles N MacLeod, Stephen G Pierce, Anthony Gachagan, David Lines, Carmelo Mineo, Jialuo Ding, Stewart Williams, Momchil Vasilev, Ehsan Mohseni, et al. Ultrasonic phased array inspection of a wire+ arc additive manufactured (waam) sample with intentionally embedded defects. *Additive Manufacturing*, 29:100806, 2019.
- [64] PA Petcher and S Dixon. Weld defect detection using ppm emat generated shear horizontal ultrasound. *Ndt & E International*, 74:58–65, 2015.
- [65] Guanghao Zhou, Guocheng Xu, Xiaopeng Gu, Jing Liu, Yukuo Tian, and Liang Zhou. Simulation and experimental study on the quality evaluation of laser welds based on ultrasonic test. *The International Journal of Advanced Manufacturing Technology*, 93(9):3897–3906, 2017.
- [66] John Stavridis, Alexios Papacharalampopoulos, and Panagiotis Stavropoulos. Quality assessment in laser welding: a critical review. *The International Journal of Advanced Manufacturing Technology*, 94(5-8):1825–1847, 2018.
- [67] Grégoire Pichenot and Thierry Sollier. Eddy current modelling for nondestructive testing. *Journal of Nondestructive Testing*, 8(6):1–5, 2003.
- [68] B Rogé, A Fahr, JSR Giguere, and KI McRae. Nondestructive measurement of porosity in thermal barrier coatings. *Journal of thermal spray technology*, 12(4):530–535, 2003.
- [69] Yao Chen, Xing Peng, Lingbao Kong, Guangxi Dong, Afaf Remani, and Richard Leach. Defect inspection technologies for additive manufacturing. *International Journal of Extreme Manufacturing*, 3(2):022002, 2021.

- [70] G Saeed and YM Zhang. Weld pool surface depth measurement using a calibrated camera and structured light. *Measurement Science and Technology*, 18(8):2570, 2007.
- [71] Iker Garmendia, Josu Leunda, Joseba Pujana, and Aitzol Lamikiz. In-process height control during laser metal deposition based on structured light 3d scanning. *Procedia Cirp*, 68:375–380, 2018.
- [72] Manuel Rodriguez-Martin, Pablo Rodriguez-Gonzalvez, Diego Gonzalez-Aguilera, and Jesus Fernandez-Hernandez. Feasibility study of a structured light system applied to welding inspection based on articulated coordinate measure machine data. *IEEE Sensors Journal*, 17(13):4217–4224, 2017.
- [73] Xiao Zhang, Weijun Shen, Vignesh Suresh, Jakob Hamilton, Li-Hsin Yeh, Xuepeng Jiang, Zhan Zhang, Qing Li, Beiwen Li, Iris V Rivero, et al. In-situ monitoring of direct energy deposition via structured light system and its application in remanufacturing industry. -, 2021.
- [74] DY You, XD Gao, and S Katayama. Review of laser welding monitoring. *Science and technology of welding and joining*, 19(3):181–201, 2014.
- [75] Jiaqing Shao and Yong Yan. Review of techniques for on-line monitoring and inspection of laser welding. In *Journal of Physics: Conference Series*, volume 15, page 101. IOP Publishing, 2005.
- [76] Justin Whiting, Adam Springer, and Federico Sciammarella. Real-time acoustic emission monitoring of powder mass flow rate for directed energy deposition. *Additive manufacturing*, 23:312–318, 2018.
- [77] Maria Strantza, Dimitrios G Aggelis, Dieter De Baere, Patrick Guillaume, and Danny Van Hemelrijck. Evaluation of shm system produced by additive manufacturing via acoustic emission and other ndt methods. *Sensors*, 15(10):26709–26725, 2015.
- [78] Maria Strantza, Danny Van Hemelrijck, Patrick Guillaume, and Dimitrios G Aggelis. Acoustic emission monitoring of crack propagation in additively manufactured and conventional titanium components. *Mechanics Research Communications*, 84:8–13, 2017.
- [79] Lucas W Koester, Hossein Taheri, Timothy A Bigelow, Leonard J Bond, and Eric J Faierson. In-situ acoustic signature monitoring in additive manufacturing processes. In *AIP Conference Proceedings*, volume 1949, page 020006. AIP Publishing LLC, 2018.

- [80] Guijun Bi, Andres Gasser, Konrad Wissenbach, Alexander Drenker, and Reinhart Poprawe. Identification and qualification of temperature signal for monitoring and control in laser cladding. *Optics and lasers in engineering*, 44(12):1348–1359, 2006.
- [81] Gilbert Held. *Introduction to light emitting diode technology and applications*. CRC press, 2016.
- [82] Young Whan Park, Hyunsung Park, Sehun Rhee, and Munjin Kang. Real time estimation of co2 laser weld quality for automotive industry. *Optics & Laser Technology*, 34(2):135–142, 2002.
- [83] Gustavo Tapia and Alaa Elwany. A review on process monitoring and control in metal-based additive manufacturing. *Journal of Manufacturing Science and Engineering*, 136(6), 2014.
- [84] Optecks. The detectors used within spectroscopy. <https://www.optecks.com/Portal/index.php/knowledge-center/spectroscopy-root/spect7>, 2021.
- [85] Cheol-Hee Kim and Do-Chang Ahn. Coaxial monitoring of keyhole during yb: Yag laser welding. *Optics & Laser Technology*, 44(6):1874–1880, 2012.
- [86] Maurizio Motta, Ali Gökhan Demir, and Barbara Previtali. High-speed imaging and process characterization of coaxial laser metal wire deposition. *Additive Manufacturing*, 22:497–507, 2018.
- [87] Joseph N Zalameda, Eric R Burke, Robert A Hafley, Karen M Taminger, Christopher S Domack, Amy Brewer, and Richard E Martin. Thermal imaging for assessment of electron-beam freeform fabrication (ebf3) additive manufacturing deposits. In *Thermosense: Thermal Infrared Applications XXXV*, volume 8705, page 87050M. International Society for Optics and Photonics, 2013.
- [88] Florian Wirth, Samuel Arpagaus, and Konrad Wegener. Analysis of melt pool dynamics in laser cladding and direct metal deposition by automated high-speed camera image evaluation. *Additive Manufacturing*, 21:369–382, 2018.
- [89] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

- [90] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.
- [91] Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*, 2020.
- [92] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- [93] Kevin Scott. Microsoft teams up with openai to exclusively license gpt-3 language model. *Official Microsoft Blog*, 2020.
- [94] Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020.
- [95] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.
- [96] Dimensional Research. Artificial intelligence and machine learning projects are obstructed by data issues. <https://cdn2.hubspot.net/hubfs/3971219/Survey%20Assets%201905/Dimensional%20Research%20Machine%20Learning%20PPT%20Report%20FINAL.pdf>, 2019.
- [97] Shilpa Arora, Eric Nyberg, and Carolyn Rose. Estimating annotation cost for active learning in a multi-annotator environment. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, pages 18–26, 2009.
- [98] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.
- [99] Reena Pagare and Anita Shinde. A study on image annotation techniques. *International Journal of Computer Applications*, 37(6):42–45, 2012.
- [100] Pei-Yun Hsueh, Prem Melville, and Vikas Sindhwani. Data quality from crowdsourcing: a study of annotation selection criteria. In *Proceedings of the NAACL*

- HLT 2009 workshop on active learning for natural language processing*, pages 27–35, 2009.
- [101] Jennifer M Durden, Brian J Bett, Timm Schoening, Kirsty J Morris, Tim W Nattkemper, and Henry A Ruhl. Comparison of image annotation data generated by multiple investigators for benthic ecology. *Marine Ecology Progress Series*, 552:61–70, 2016.
 - [102] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.
 - [103] Peter Lloyd-Sherlock, Lucas Sempe, Martin McKee, and Aravinda Guntupalli. Problems of data availability and quality for covid-19 and older people in low-and middle-income countries. *The Gerontologist*, 61(2):141–144, 2021.
 - [104] Domingos P. A few useful things to know about machine learning. *Comm. of the ACM*, 2012.
 - [105] Michael A Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4845–4854, 2019.
 - [106] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
 - [107] Adarsh Subbaswamy, Roy Adams, and Suchi Saria. Evaluating model robustness and stability to dataset shift. In *International Conference on Artificial Intelligence and Statistics*, pages 2611–2619. PMLR, 2021.
 - [108] Xiangxin Zhu, Carl Vondrick, Deva Ramanan, and Charless C Fowlkes. Do we need more training data or better models for object detection?. In *BMVC*, volume 3. Citeseer, 2012.
 - [109] Vighnesh Birodkar, Hossein Mobahi, and Samy Bengio. Semantic redundancies in image-classification datasets: The 10% you don’t need. *arXiv preprint arXiv:1901.11409*, 2019.

- [110] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9226–9235, 2019.
- [111] Wenyuan Cui, Yunlu Zhang, Xinchang Zhang, Lan Li, and Frank Liou. Metal additive manufacturing parts inspection using convolutional neural network. *Applied Sciences*, 10(2):545, 2020.
- [112] Haixing Zhu, Weimin Ge, and Zhenzhong Liu. Deep learning-based classification of weld surface defects. *Applied Sciences*, 9(16):3312, 2019.
- [113] Jun Sun, Chao Li, Xiao-Jun Wu, Vasile Palade, and Wei Fang. An effective method of weld defect detection and classification based on machine vision. *IEEE Transactions on Industrial Informatics*, 15(12):6322–6333, 2019.
- [114] Paolo Sassi, Paolo Tripicchio, and Carlo Alberto Avizzano. A smart monitoring system for automatic welding defect detection. *IEEE Transactions on Industrial Electronics*, 66(12):9641–9650, 2019.
- [115] Wenhui Hou, Ye Wei, Yi Jin, and Changan Zhu. Deep features based on a dcnn model for classifying imbalanced weld flaw types. *Measurement*, 131:482–489, 2019.
- [116] Benzhi Chen, Zhihong Fang, Yong Xia, Ling Zhang, Yijie Huang, and Lisheng Wang. Accurate defect detection via sparsity reconstruction for weld radiographs. *NDT & E International*, 94:62–69, 2018.
- [117] Neury Boaretto and Tania Mezzadri Centeno. Automated detection of welding defects in pipelines from radiographic images dwdi. *Ndt & E International*, 86:7–13, 2017.
- [118] Mohamed S El-Tokhy and Imbaby I Mahmoud. Classification of welding flaws in gamma radiography images based on multi-scale wavelet packet feature extraction using support vector machine. *Journal of Nondestructive Evaluation*, 34(4):1–17, 2015.
- [119] Juan Zapata, Rafael Vilar, and Ramón Ruiz. Performance evaluation of an automatic inspection system of weld defects in radiographic images based on neuro-classifiers. *Expert Systems with Applications*, 38(7):8812–8824, 2011.
- [120] Ioannis Valavanis and Dimitrios Kosmopoulos. Multiclass defect detection and classification in weld radiographic images using geometric and texture features. *Expert Systems with Applications*, 37(12):7606–7614, 2010.

- [121] Hossein Taheri, Lucas W Koester, Timothy A Bigelow, Eric J Faierson, and Leonard J Bond. In situ additive manufacturing process monitoring with an acoustic technique: clustering performance evaluation using k-means algorithm. *Journal of Manufacturing Science and Engineering*, 141(4), 2019.
- [122] Zhifen Zhang, Guangrui Wen, and Shanben Chen. Audible sound-based intelligent evaluation for aluminum alloy in robotic pulsed gtaw: Mechanism, feature selection, and defect detection. *IEEE Transactions on Industrial Informatics*, 14(7):2973–2983, 2017.
- [123] Na Lv, Yanling Xu, Sichen Li, Xinwen Yu, and Shanben Chen. Automated control of welding penetration based on audio sensing technology. *Journal of Materials Processing Technology*, 250:81–98, 2017.
- [124] Yiming Huang, Shuaishuai Hou, Shufeng Xu, Shengbin Zhao, Lijun Yang, and Zhifen Zhang. Emd-pnn based welding defects detection using laser-induced plasma electrical signals. *Journal of Manufacturing Processes*, 45:642–651, 2019.
- [125] Zhehao Zhang, Bin Li, Weifeng Zhang, Rundong Lu, Satoshi Wada, and Yi Zhang. Real-time penetration state monitoring using convolutional neural network for laser welding of tailor rolled blanks. *Journal of Manufacturing Systems*, 54:348–360, 2020.
- [126] Tianyuan Liu, Jinsong Bao, Junliang Wang, and Yiming Zhang. A hybrid cnn-lstm algorithm for online defect recognition of co2 welding. *Sensors*, 18(12):4369, 2018.
- [127] Juequan Chen, Teng Wang, Xiangdong Gao, and Li Wei. Real-time monitoring of high-power disk laser welding based on support vector machine. *Computers in Industry*, 94:75–81, 2018.
- [128] Yunhe Feng, Zongyao Chen, Dali Wang, Jian Chen, and Zhili Feng. Deepwelding: A deep learning enhanced approach to gtaw using multisource sensing images. *IEEE Transactions on Industrial Informatics*, 16(1):465–474, 2019.
- [129] Zhifen Zhang, Guangrui Wen, and Shanben Chen. Weld image deep learning-based on-line defects detection using convolutional neural networks for al alloy in robotic arc welding. *Journal of Manufacturing Processes*, 45:208–216, 2019.
- [130] I-Hsi Kao, Ya-Wen Hsu, Yi Horng Lai, and Jau-Woei Perng. Laser cladding quality monitoring using coaxial image based on machine learning. *IEEE Transactions on Instrumentation and Measurement*, 2019.

- [131] Carlos Gonzalez-Val, Adrian Pallas, Veronica Panadeiro, and Alvaro Rodriguez. A convolutional approach to quality monitoring for laser manufacturing. *Journal of Intelligent Manufacturing*, 31(3):789–795, 2020.
- [132] Sergey Shevchik, Tri Le-Quang, Bastian Meylan, Farzad Vakili Farahani, Margie P Olbinado, Alexander Rack, Giulio Masinelli, Christian Leinenbach, and Kilian Wasmer. Supervised deep learning for real-time quality monitoring of laser welding with x-ray radiographic guidance. *Scientific Reports*, 10(1):1–12, 2020.
- [133] Yanxi Zhang, Deyong You, Xiangdong Gao, Nanfeng Zhang, and Perry P Gao. Welding defects detection based on deep learning with multiple optical sensors during disk laser welding of thick plates. *Journal of Manufacturing Systems*, 51:87–94, 2019.
- [134] Mohammad Montazeri, Abdalla R Nassar, Christopher B Stutzman, and Prahalada Rao. Heterogeneous sensor-based condition monitoring in directed energy deposition. *Additive Manufacturing*, 30:100916, 2019.
- [135] Di Wu, Yiming Huang, Huabin Chen, Yinshui He, and Shanben Chen. Vppaw penetration monitoring based on fusion of visual and acoustic signals using t-sne and dbn model. *Materials & Design*, 123:1–14, 2017.
- [136] Guiqian Liu, Xiangdong Gao, Deyong You, and Nanfeng Zhang. Prediction of high power laser welding status based on pca and svm classification of multiple sensors. *Journal of Intelligent Manufacturing*, 30(2):821–832, 2019.
- [137] Johannes Günther, Patrick M Pilarski, Gerhard Helfrich, Hao Shen, and Klaus Diepold. Intelligent laser welding through representation, prediction, and control learning: An architecture with deep neural networks and reinforcement learning. *Mechatronics*, 34:1–11, 2016.
- [138] Zhifen Zhang, Huabin Chen, Yanling Xu, Jiyong Zhong, Na Lv, and Shanben Chen. Multisensor-based real-time quality monitoring by means of feature extraction, selection and modeling for al alloy in arc welding. *Mechanical Systems and Signal Processing*, 60:151–165, 2015.
- [139] Deyong You, Xiangdong Gao, and Seiji Katayama. Wpd-pca-based laser welding process monitoring and defects diagnosis by using fnn and svm. *IEEE Transactions on Industrial Electronics*, 62(1):628–636, 2014.
- [140] Bin Zhang, Shunyu Liu, and Yung C Shin. In-process monitoring of porosity during laser additive manufacturing process. *Additive Manufacturing*, 28:497–505, 2019.

- [141] Mojtaba Khanzadeh, Sudipta Chowdhury, Mohammad Marufuzzaman, Mark A Tschopp, and Linkan Bian. Porosity prediction: Supervised-learning of thermal history for direct laser deposition. *Journal of manufacturing systems*, 47:69–82, 2018.
- [142] Mojtaba Khanzadeh, Sudipta Chowdhury, Mark A Tschopp, Haley R Doude, Mohammad Marufuzzaman, and Linkan Bian. In-situ monitoring of melt pool images for porosity prediction in directed energy deposition processes. *IISE Transactions*, 51(5):437–455, 2019.
- [143] Seyyed Hadi Seifi, Wenmeng Tian, Aref Yadollahi, Haley Doude, and Linkan Bian. In-situ fatigue prediction of direct laser deposition parts based on thermal profile. In *ASME 2019 international mechanical engineering congress and exposition*. American Society of Mechanical Engineers Digital Collection, 2019.
- [144] Qi Tian, Shenghan Guo, Erika Melder, Linkan Bian, Weihong Guo, et al. Deep learning-based data fusion method for in situ porosity detection in laser-based additive manufacturing. *Journal of Manufacturing Science and Engineering*, 143(4), 2021.
- [145] Gijs JJ van Houtum and Mihaela L Vlasea. Active learning via adaptive weighted uncertainty sampling applied to additive manufacturing. *Additive Manufacturing*, 48:102411, 2021.
- [146] Robert Sampson, Robert Lancaster, Mark Sutcliffe, David Carswell, Carl Hauser, and Josh Barras. An improved methodology of melt pool monitoring of direct energy deposition processes. *Optics & Laser Technology*, 127:106194, 2020.
- [147] Brian T Gibson, Yashwanth Kumar Bandari, Bradley S Richardson, William C Henry, Emma J Vetland, Tayler W Sundermann, and Lonnie J Love. Melt pool size control through multiple closed-loop modalities in laser-wire directed energy deposition of ti-6al-4v. *Additive Manufacturing*, 32:100993, 2020.
- [148] C Knaak, G Kolter, F Schulze, M Kröger, and P Abels. Deep learning-based semantic segmentation for in-process monitoring in laser welding applications. In *Applications of Machine Learning*, volume 11139, page 1113905. International Society for Optics and Photonics, 2019.
- [149] Burr Settles, Mark Craven, and Lewis Friedland. Active learning with real annotation costs. In *Proceedings of the NIPS workshop on cost-sensitive learning*, pages 1–10. Vancouver, CA:, 2008.

- [150] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.
- [151] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer, 1994.
- [152] Punit Kumar and Atul Gupta. Active learning query strategies for classification, regression, and clustering: A survey. *Journal of Computer Science and Technology*, 35(4):913–945, 2020.
- [153] Yifan Fu, Xingquan Zhu, and Bin Li. A survey on instance selection for active learning. *Knowledge and information systems*, 35(2):249–283, 2013.
- [154] Yazhou Yang and Marco Loog. A benchmark and comparison of active learning for logistic regression. *Pattern Recognition*, 83:401–415, 2018.
- [155] Davi Pereira-Santos, Ricardo Bastos Cavalcante Prudêncio, and André CPLF de Carvalho. Empirical investigation of active learning strategies. *Neurocomputing*, 326:15–27, 2019.
- [156] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.
- [157] Ksenia Konyushkova, Raphael Sznitman, and Pascal Fua. Discovering general-purpose active learning strategies. *arXiv preprint arXiv:1810.04114*, 2018.
- [158] Arantxa Casanova, Pedro O Pinheiro, Negar Rostamzadeh, and Christopher J Pal. Reinforced active learning for image segmentation. *arXiv preprint arXiv:2002.06583*, 2020.
- [159] Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2372–2379. IEEE, 2009.
- [160] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.
- [161] Seyda Ertekin, Jian Huang, Leon Bottou, and Lee Giles. Learning on the border: active learning in imbalanced data classification. In *Proceedings of the sixteenth*

- ACM conference on Conference on information and knowledge management*, pages 127–136, 2007.
- [162] H Sebastian Seung, Manfred Oppel, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294, 1992.
 - [163] Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, pages 441–448, 2001.
 - [164] Alex Holub, Pietro Perona, and Michael C Burl. Entropy-based active learning for object recognition. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008.
 - [165] Yazhou Yang and Marco Loog. A variance maximization criterion for active learning. *Pattern Recognition*, 78:358–370, 2018.
 - [166] Priscila TM Saito, Celso TN Suzuki, Jancarlo F Gomes, Pedro J de Rezende, and Alexandre X Falcão. Robust active learning for the diagnosis of parasites. *Pattern Recognition*, 48(11):3572–3583, 2015.
 - [167] Yanshan Xiao, Zheng Chang, and Bo Liu. An efficient active learning method for multi-task learning. *Knowledge-Based Systems*, 190:105137, 2020.
 - [168] Klaus Brinker. Incorporating diversity in active learning with support vector machines. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 59–66, 2003.
 - [169] Thiago NC Cardoso, Rodrigo M Silva, Sérgio Canuto, Mirella M Moro, and Marcos A Gonçalves. Ranked batch-mode active learning. *Information Sciences*, 379:313–337, 2017.
 - [170] Zuobing Xu, Ram Akella, and Yi Zhang. Incorporating diversity and density in active learning for relevance feedback. In *European Conference on Information Retrieval*, pages 246–257. Springer, 2007.
 - [171] Tianxu He, Shukui Zhang, Jie Xin, Pengpeng Zhao, Jian Wu, Xuefeng Xian, Chunhua Li, and Zhiming Cui. An active learning approach with uncertainty, representativeness, and diversity. *The Scientific World Journal*, 2014, 2014.

- [172] Seho Kee, Enrique Del Castillo, and George Runger. Query-by-committee improvement with diversity and density in batch active learning. *Information Sciences*, 454:401–418, 2018.
- [173] Andrew I Schein and Lyle H Ungar. Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265, 2007.
- [174] Daria Zotova, Aneta Lisowska, Owen Anderson, Vismantas Dilys, and Alison O’Neil. Comparison of active learning strategies applied to lung nodule segmentation in ct scans. In *Large-Scale Annotation of Biomedical Data and Expert Label Synthesis and Hardware Aware Learning for Medical Imaging and Computer Assisted Intervention*, pages 3–12. Springer, 2019.
- [175] Maytal Saar-Tsechansky and Foster Provost. Active sampling for class probability estimation and ranking. *Machine learning*, 54(2):153–178, 2004.
- [176] Thomas Osugi, Deng Kim, and Stephen Scott. Balancing exploration and exploitation: A new algorithm for active machine learning. In *Fifth IEEE International Conference on Data Mining (ICDM’05)*, pages 8–pp. IEEE, 2005.
- [177] Djallel Bouneffouf. Exponentiated gradient exploration for active learning. *Computers*, 5(1):1, 2016.
- [178] Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013.
- [179] Wenhong Li, Yunhong Ji, Jing Wu, and Jiayou Wang. A modified welding image feature extraction algorithm for rotating arc narrow gap mag welding. *Industrial Robot: An International Journal*, 2015.
- [180] Masiyang Luo and Yung C Shin. Vision-based weld pool boundary extraction and width measurement during keyhole fiber laser welding. *Optics and Lasers in Engineering*, 64:59–70, 2015.
- [181] Di Wu, Huabin Chen, Yinshui He, Shuo Song, Tao Lin, and Shanben Chen. A prediction model for keyhole geometry and acoustic signatures during variable polarity plasma arc welding based on extreme learning machine. *Sensor Review*, 2016.
- [182] Lijun Song, Fanghua Wang, Simeng Li, and Xu Han. Phase congruency melt pool edge extraction for laser additive manufacturing. *Journal of Materials Processing Technology*, 250:261–269, 2017.

- [183] Mark Jager and Fred A Hamprecht. Principal component imagery for the quality monitoring of dynamic laser welding processes. *IEEE Transactions on Industrial Electronics*, 56(4):1307–1313, 2008.
- [184] D Kelley. *Introduction to probability*. Macmillan Publishing Company, London, 1994.
- [185] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [186] Cheng Deng, Xianglong Liu, Chao Li, and Dacheng Tao. Active multi-kernel domain adaptation for hyperspectral image classification. *Pattern Recognition*, 77:306–315, 2018.
- [187] EEO Ishida, R Beck, S González-Gaitán, RS de Souza, A Krone-Martins, JW Barrett, N Kennamer, R Vilalta, JM Burgess, B Quint, et al. Optimizing spectroscopic follow-up strategies for supernova photometric classification with active learning. *Monthly Notices of the Royal Astronomical Society*, 483(1):2–18, 2019.
- [188] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [189] Ronghua Luo and Xiang Wang. Batch active learning with two-stage sampling. *IEEE Access*, 8:46518–46528, 2020.
- [190] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [191] Syed AM Tofail, Elias P Koumoulos, Amit Bandyopadhyay, Susmita Bose, Lisa O’Donoghue, and Costas Charitidis. Additive manufacturing: scientific and technological challenges, market uptake and opportunities. *Materials today*, 21(1):22–37, 2018.
- [192] Ronan McCann, Muhannad A Obeidi, Cian Hughes, Éanna McCarthy, Darragh S Egan, Rajani K Vijayaraghavan, Ajey M Joshi, Victor Acinas Garzon, Denis P Dowling, Patrick J McNally, et al. In-situ sensing, process monitoring and machine control in laser powder bed fusion: A review. *Additive Manufacturing*, 45:102058, 2021.

- [193] Youssef AbouelNour and Nikhil Gupta. In-situ monitoring of sub-surface and internal defects in additive manufacturing: A review. *Materials & Design*, page 111063, 2022.
- [194] Jian Qin, Fu Hu, Ying Liu, Paul Witherell, Charlie CL Wang, David W Rosen, Timothy W Simpson, Yan Lu, and Qian Tang. Research and application of machine learning for additive manufacturing. *Additive Manufacturing*, 52:102691, 2022.
- [195] Yanzhou Fu, Austin RJ Downey, Lang Yuan, Tianyu Zhang, Avery Pratt, and Yunusa Balogun. Machine learning algorithms for defect detection in metal laser-based additive manufacturing: A review. *Journal of Manufacturing Processes*, 75:693–710, 2022.
- [196] M Bhuvanesh Kumar and P Sathiya. Methods and materials for additive manufacturing: A critical review on advancements and challenges. *Thin-Walled Structures*, 159:107228, 2021.
- [197] Alexander Oleff, Benjamin Küster, Malte Stonis, and Ludger Overmeyer. Process monitoring for material extrusion additive manufacturing: a state-of-the-art review. *Progress in Additive Manufacturing*, pages 1–26, 2021.
- [198] Martin Baumers and Matthias Holweg. On the economics of additive manufacturing: Experimental findings. *Journal of Operations Management*, 65(8):794–809, 2019.
- [199] Yifan Tang, M Rahmani Dehaghani, and G Gary Wang. Review of transfer learning in modeling additive manufacturing processes. *Additive Manufacturing*, 61:103357, 2023.
- [200] Deniz Sera Ertay, Mohamed A Naiel, Mihaela Vlasea, and Paul Fieguth. Process performance evaluation and classification via in-situ melt pool monitoring in directed energy deposition. *CIRP Journal of Manufacturing Science and Technology*, 35:298–314, 2021.
- [201] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [202] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, et al. Symbolic discovery of optimization algorithms. *arXiv preprint arXiv:2302.06675*, 2023.

- [203] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4927, 2019.
- [204] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [205] Simone Bianco, Remi Cadene, Luigi Celona, and Paolo Napoletano. Benchmark analysis of representative deep neural network architectures. *IEEE access*, 6:64270–64277, 2018.
- [206] James C Haley, Julie M Schoenung, and Enrique J Lavernia. Observations of particle-melt pool impact events in directed energy deposition. *Additive Manufacturing*, 22:368–374, 2018.
- [207] Shi-Hao Li, Punit Kumar, Shubham Chandra, and Upadrasta Ramamurty. Directed energy deposition of metals: processing, microstructures, and mechanical properties. *International Materials Reviews*, 68(6):605–647, 2023.
- [208] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [209] Hao Guan and Mingxia Liu. Domain adaptation for medical image analysis: a survey. *IEEE Transactions on Biomedical Engineering*, 69(3):1173–1185, 2021.
- [210] Ahmet Ali Süzen, Burhan Duman, and Betül Şen. Benchmark analysis of jetson tx2, jetson nano and raspberry pi using deep-cnn. In *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pages 1–5. IEEE, 2020.
- [211] Olivia Wiles, Sven Goyal, Florian Stimberg, Sylvestre Alvisé-Rebuffi, Ira Ktena, Krishnamurthy Dvijotham, and Taylan Cemgil. A fine-grained analysis on distribution shift. *arXiv preprint arXiv:2110.11328*, 2021.

- [212] Zhuo Wang, Wenhua Yang, Qingyang Liu, Yingjie Zhao, Pengwei Liu, Dazhong Wu, Mihaela Banu, and Lei Chen. Data-driven modeling of process, structure and property in additive manufacturing: A review and future directions. *Journal of Manufacturing Processes*, 77:13–31, 2022.
- [213] Tuğrul Özel, Hamed Shokri, and Raphaël Loizeau. A review on wire-fed directed energy deposition based metal additive manufacturing. *Journal of Manufacturing and Materials Processing*, 7(1):45, 2023.
- [214] Weiwei Liu, Tandong Wang, Bingjun Liu, Wanyang Li, Guangda Hu, and Zhenxin Lyu. Research of on-line monitoring technology and control strategy for laser-directed energy deposition: a review. *The International Journal of Advanced Manufacturing Technology*, pages 1–28, 2024.
- [215] Reza Asadi, Antoine Queguineur, Olli Wiikinkoski, Hossein Mokhtarian, Tommi Aihkialo, Alejandro Revuelta, and Iñigo Flores Ituarte. Process monitoring by deep neural networks in directed energy deposition: Cnn-based detection, segmentation, and statistical analysis of melt pools. *Robotics and Computer-Integrated Manufacturing*, 87:102710, 2024.
- [216] Lucy Ellen Lwakatare, Aiswarya Raj, Ivica Crnkovic, Jan Bosch, and Helena Holmström Olsson. Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions. *Information and software technology*, 127:106368, 2020.
- [217] Mingqi Gao, Feng Zheng, James JQ Yu, Caifeng Shan, Guiguang Ding, and Jungong Han. Deep learning for video object segmentation: a review. *Artificial Intelligence Review*, 56(1):457–531, 2023.
- [218] Ho Kei Cheng, Yu-Wing Tai, and Chi-Keung Tang. Rethinking space-time networks with improved memory coverage for efficient video object segmentation. *Advances in Neural Information Processing Systems*, 34:11781–11794, 2021.
- [219] Haochen Wang, Xiaolong Jiang, Haibing Ren, Yao Hu, and Song Bai. Swiftnet: Real-time video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1296–1305, 2021.
- [220] Roy Miles, Mehmet Kerim Yucel, Bruno Manganelli, and Albert Saà-Garriga. Mobilevos: Real-time video object segmentation contrastive learning meets knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10480–10490, 2023.

- [221] Evangelos Skartados, Konstantinos Georgiadis, M Kerim Yucel, Koskinas Ioannis, Armando Domi, Anastasios Drosou, Bruno Manganelli, and Albert Saà-Garriga. Trickvos: A bag of tricks for video object segmentation. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 2430–2434. IEEE, 2023.
- [222] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016.
- [223] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019.
- [224] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.
- [225] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.
- [226] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [227] Alaa Tharwat and Wolfram Schenck. A survey on active learning: State-of-the-art, practical challenges and research directions. *Mathematics*, 11(4):820, 2023.
- [228] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.
- [229] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

APPENDICES

Appendix A

Annotation protocol

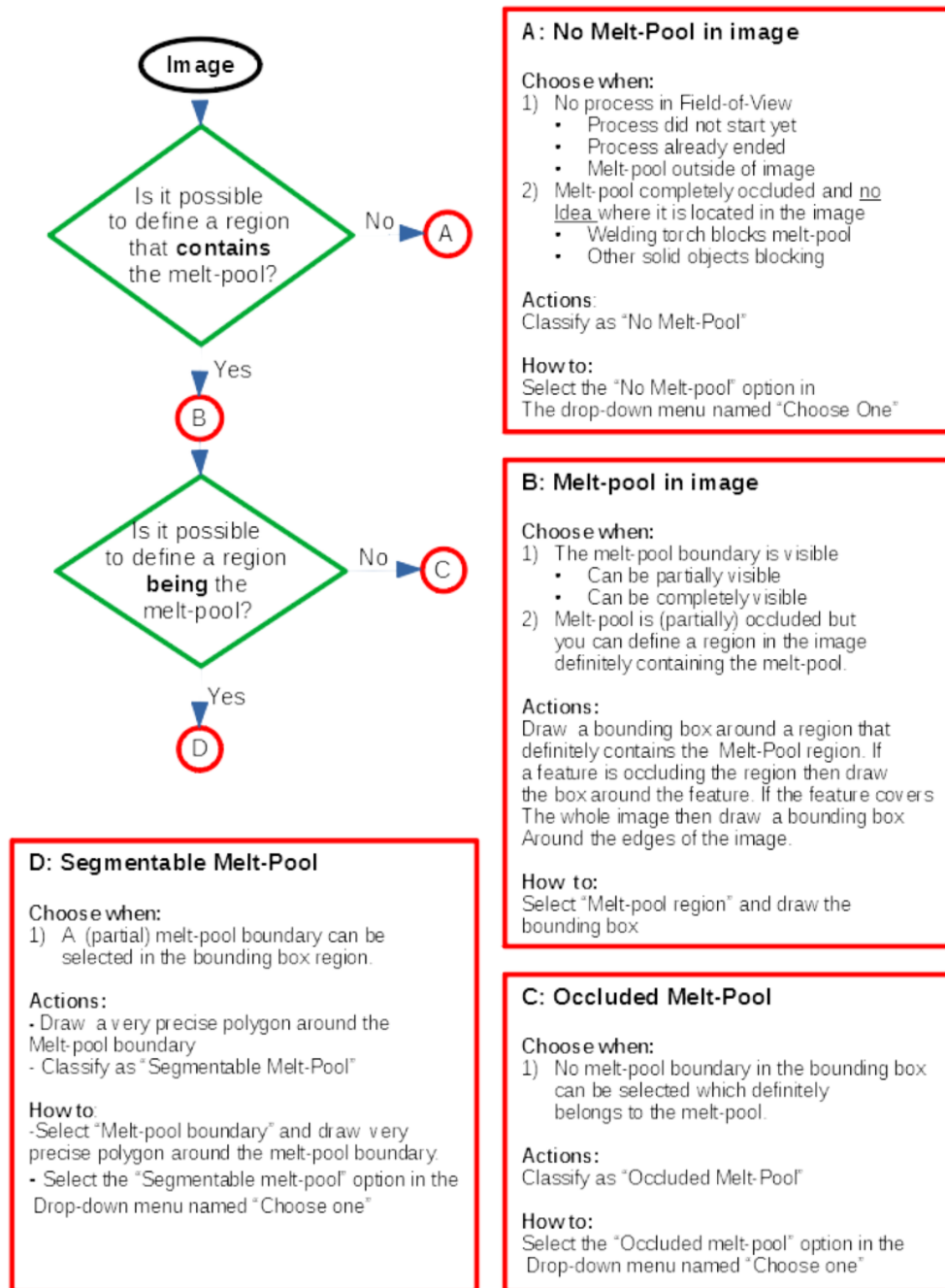


Figure A.1: Annotation protocol used to create DED datasets described in Chapters 3, 4, and 5.

Appendix B

Active learning

Table B.1: Pool-based Batch-mode Active Learning (PBAL)

Input	: U - Pool of unlabeled data instances L - Pool of labeled data instances C - Set of unique class labels in $U \cup L$ m - Classification model not trained on L H - Annotators b - Batch size t - Termination condition q - Query strategy function	
Output	: $L, \bar{m}$	
1	$\bar{m} \xleftarrow{\text{train}} L, C$	// Train model on labeled data
2	while $ U > 0$ <i>and</i> t <i>is satisfied</i> do	
3	$U_B \leftarrow q(U, L, m, \bar{m}, C, b)$	// Query batch from U
4	$L_B \xleftarrow{\text{annotate}} H, C, U_B$	// Annotators provide labels
5	$L \leftarrow L \cup L_B$	// Add to labeled
6	$U \leftarrow U \setminus U_B$	// Remove from unlabeled
7	$m \leftarrow \bar{m}$	// Set current to previous model
8	$\bar{m} \xleftarrow{\text{train}} L, C$	// Re-train model

Table B.2: 28 open-source datasets from the OpenML database and 8 datasets from 8 different directed-energy-deposition (DED) processes. The dataset name (N), number of instances (I), features (F) and unique classes (C) per dataset are presented on the left. F1-macro classification performance, mean and standard deviation in %, on the validation set (50%-50% train-validation split) on the right over 1000 repeated random dataset splits

Dataset				Classifier			
N	I	F	C	GNB	LR	RF	SVM
transplant	131	3	2	85.3±1.4	86.2±1.4	85.0±1.6	86.1±1.4
iris	150	4	3	91.3±2.5	90.8±2.7	90.9±2.8	90.8±2.5
tae	151	5	3	89.2±2.2	91.3±2.0	91.7±2.3	92.5±1.9
corral	160	6	2	79.7±2.0	89.8±1.6	91.4±1.7	90.5±1.6
servo	167	4	2	95.6±1.4	96.0±1.1	97.9±1.1	97.7±1.1
wine	178	13	3	70.9±2.2	73.9±2.3	81.3±2.2	77.5±2.9
<u>DED-a</u>	181	16	3	76.2±2.1	83.0±2.5	87.7±2.2	88.5±2.2
flags	194	28	2	81.2±6.5	97.0±2.7	86.9±7.7	86.2±7.0
sonar	208	60	2	86.2±3.5	80.7±4.4	96.8±2.1	87.9±3.8
cpu	209	7	2	90.8±6.9	90.8±1.0	69.0±12.9	90.8±1.0
seeds	210	7	3	79.3±3.0	92.8±2.8	97.6±2.2	95.9±1.9
glass	214	9	2	74.3±2.3	85.5±1.9	85.9±1.9	86.0±1.9
bodyfat	252	14	2	79.7±2.7	78.7±2.7	77.8±2.8	78.7±2.7
heart-statlog	270	13	2	84.7±4.7	88.1±3.2	96.5±3.3	87.6±3.3
cleve	303	13	2	93.0±2.5	86.8±5.0	94.1±3.2	96.9±2.4
solar-flare	315	12	5	65.9±9.7	68.2±1.8	66.5±1.9	68.1±1.7
ecoli	336	7	2	72.0±1.9	72.5±1.9	69.2±2.1	72.9±1.8
ionosphere	351	34	2	72.9±11.6	95.6±1.2	94.5±1.6	96.3±1.1
vinnie	380	2	2	60.3±3.9	70.0±4.2	63.6±4.4	68.0±4.6
<u>DED-b</u>	412	16	3	57.1±5.1	49.3±5.2	75.8±4.3	52.0±4.5
<u>DED-c</u>	451	16	3	83.2±2.6	82.6±2.5	78.7±2.8	82.6±2.8
boston	506	13	2	86.9±3.0	85.2±2.3	91.3±1.9	85.2±2.7
wdbc	569	30	2	95.1±2.0	89.2±3.5	94.6±2.0	93.0±2.9
<u>DED-d</u>	607	16	3	90.3±2.3	91.8±2.2	90.4±2.8	94.1±2.0
strikes	625	6	2	90.8±2.4	87.2±4.5	87.5±3.8	90.3±2.7
balance-scale	625	4	3	51.4±7.2	68.6±5.6	67.7±3.9	69.5±6.3
<u>DED-e</u>	639	16	3	67.8±4.9	75.4±3.6	76.4±4.1	74.1±3.9
Australian	690	14	2	70.7±1.8	80.8±1.8	95.7±0.9	84.6±1.3
<u>DED-f</u>	694	16	3	60.2±2.8	60.1±3.1	93.9±2.1	61.8±3.1
diabetes	768	8	2	47.8±6.8	49.5±5.1	52.9±5.3	49.2±5.3
vehicle	846	18	4	96.0±2.1	96.7±2.3	97.9±2.2	98.7±1.6
<u>DED-g</u>	900	16	3	42.1±3.9	71.8±1.9	72.9±1.8	76.2±1.6
stock	950	9	2	84.3±1.8	83.7±2.1	82.1±2.6	83.8±2.1
vowel	990	12	2	84.4±2.4	85.7±3.0	95.5±2.3	90.2±2.0
credit-g	1000	20	2	92.8±1.5	96.2±1.0	94.6±1.3	96.6±0.9
<u>DED-h</u>	1078	16	3	97.0±1.7	97.9±1.4	95.9±2.3	97.5±1.5

Table B.3: Active learning simulation procedure

Input	: n - Nr of repeated simulations
	M - Set of classifiers
	Q - Set of query strategies
	B - Set of batch sizes
	D - Set of pre-annotated (labeled) datasets
	p - Classifier performance scoring function
Output	: R - Set of simulation results
1	$R \leftarrow \emptyset$ // Empty simulations results
2	$V : x \rightarrow y$ // Virtual annotator
3	$T \leftarrow \text{True}$ // Perform PBAL until all labeled
4	for L <i>in</i> D do
5	$C \leftarrow V, L$ // Identify all unique classes in dataset
6	for $i \leftarrow 1$ <i>to</i> n do
7	$L_t, L_v \xleftarrow{\text{sample}} L, C$ // Uniform 50% train-validation split per class
8	$U_t \leftarrow \{x (y, x) \in L_t\}$ // Get unlabeled train set
9	$L_0 \xleftarrow{\text{sample}} L_t, C$ // Uniform sample 1 instance per class from train set
10	$U_0 \leftarrow \{x \in U_t x \notin L_1\}$ // Update unlabeled pool
11	for m <i>in</i> M do
12	$m_b \xleftarrow{\text{train}} L_t, C, m$ // Train benchmark model
13	$s_b \xleftarrow{\text{eval}} p(m_b, L_v)$ // Get benchmark score on validation set
14	for q <i>in</i> Q do
15	for b <i>in</i> B do
16	$L_a, m_a \leftarrow \text{PBAL}(U_t, L_t, C, m, V, b, T, q)$ // Active learning, (see B.1)
17	$\bar{L} \leftarrow \emptyset$
18	for (y, x) <i>in</i> L_a do
19	$\bar{L} \leftarrow \bar{L} \cup (y, x)$ // Update for every instance
20	$\bar{m} \xleftarrow{\text{train}} \bar{L}, C, m$ // Train model
21	$\bar{s} \xleftarrow{\text{eval}} p(\bar{m}, L_v)$ // get score on validation set
22	if $\bar{s} > s_b$ then
23	$r \leftarrow \bar{L} / L_t $ // get ratio of training needed to reach benchmark
24	$R \leftarrow R \cup \{i, d, m, q, b, r\}$ // store result
25	break

Table B.4: Active learning performance of query strategies for each dataset and . Each number represents the Area-Under-Cumulative-Curve (AUCC) of 1000 repeated simulations for each batch sizes; 1,4,16 and 64. See Figure 3.2 for more details on AUCC.

dataset	GNB								LR								RF								SVM								
	awus-c	awus-r	bee	ega	rnd	udd	us	wus	awus-c	awus-r	bee	ega	rnd	udd	us	wus	awus-c	awus-r	bee	ega	rnd	udd	us	wus	awus-c	awus-r	bee	ega	rnd	udd	us	wus	
transplant	81	80	80	79	76	78	80	80	84	82	76	79	73	80	76	80	83	81	83	82	75	85	83	83	83	83	79	79	78	70	81	79	76
iris	71	71	69	71	66	55	69	69	72	66	53	60	53	58	51	63	82	81	82	80	78	83	81	82	66	63	54	61	48	58	51	55	
tae	62	63	60	63	62	60	60	60	57	57	53	55	55	59	54	56	43	43	42	42	42	47	42	43	54	56	48	54	55	57	47	56	
corral	62	63	50	59	61	58	50	52	70	69	70	66	59	69	70	66	63	61	60	58	51	67	60	62	72	70	72	68	62	71	72	67	
servo	84	84	83	83	84	85	82	83	70	67	68	65	56	73	68	64	84	83	84	83	81	85	84	84	76	75	75	73	68	79	75	72	
wine	56	55	53	55	49	56	54	55	67	65	61	61	52	63	60	58	66	65	64	63	59	67	64	65	72	69	71	68	55	71	71	59	
DED-a	70	70	69	71	68	53	68	70	86	86	83	84	81	79	83	84	79	77	77	76	73	75	78	78	85	84	84	84	80	78	84	81	
flags	82	82	81	83	83	81	80	81	52	53	52	52	50	53	52	52	76	76	76	75	75	74	76	76	56	56	54	55	55	56	53	55	
sonar	82	81	82	82	80	80	82	82	51	52	49	50	49	51	49	51	58	57	58	57	57	59	57	57	57	57	54	55	55	57	54	56	
cpu	70	70	69	69	65	74	69	70	83	79	84	81	53	87	83	72	75	74	75	72	62	75	75	74	75	70	75	70	46	76	74	57	
seeds	71	71	66	71	70	59	67	68	83	79	74	78	68	76	73	75	71	70	70	67	61	72	70	69	74	69	76	73	52	77	75	59	
glass	88	88	87	89	89	83	86	86	92	93	91	92	94	92	91	93	54	56	52	53	53	54	52	54	89	91	86	89	92	87	85	91	
bodyfat	87	86	87	86	84	87	86	87	68	62	68	59	34	71	67	52	78	78	78	76	72	79	78	77	68	63	69	62	39	69	69	50	
heart-statlog	49	51	34	51	51	33	34	37	72	70	72	69	62	70	72	68	78	78	77	77	77	74	77	78	60	58	60	57	51	60	60	54	
cleve	56	59	41	59	62	43	40	43	67	67	66	64	62	65	66	64	77	77	76	77	76	74	76	77	60	59	59	57	56	61	59	56	
solar-flare	80	80	74	79	81	74	73	76	56	51	51	51	40	58	51	45	79	79	78	78	77	79	77	78	49	46	46	48	41	56	45	42	
ecoli	96	96	96	96	91	95	95	95	83	81	78	77	65	83	77	76	82	82	80	80	75	83	80	81	81	79	80	77	61	84	79	69	
ionosphere	68	70	63	68	68	53	63	63	53	48	53	50	38	57	52	41	70	70	68	67	62	59	67	69	40	39	40	44	41	46	40	37	
vinnie	86	87	82	86	85	84	82	82	86	85	81	82	81	80	80	82	78	77	77	78	73	73	78	78	88	87	85	86	86	84	84	86	
DED-b	70	69	67	69	64	70	67	67	84	82	82	79	62	85	82	74	78	77	77	75	65	80	77	76	76	73	74	70	51	76	74	57	
DED-c	74	77	66	78	77	74	65	66	64	60	60	56	36	64	59	48	68	66	67	63	53	70	66	66	58	56	57	54	44	59	57	49	
boston	88	89	80	90	91	78	81	81	67	65	66	58	45	65	66	56	75	74	73	72	67	74	73	73	65	64	64	61	52	61	64	57	
wdbc	89	88	90	89	84	92	91	90	85	83	82	80	44	83	81	74	84	83	83	81	70	84	83	82	82	79	81	77	53	83	81	68	
DED-d	68	68	61	65	62	78	59	64	76	75	71	70	47	77	70	66	78	76	77	72	51	74	77	75	75	72	75	69	35	65	75	44	
balance-scale	66	66	63	58	37	60	63	57	84	84	80	80	80	81	80	83	73	73	72	71	68	72	71	73	78	81	74	74	78	75	73	80	
strikes	97	97	97	97	97	98	97	97	81	82	79	82	80	79	79	81	61	59	59	53	38	62	59	55	72	74	69	75	73	69	68	74	
DED-e	91	92	86	92	93	90	87	87	93	92	91	90	83	94	90	87	79	76	78	75	64	80	78	75	85	83	82	81	69	84	81	71	
Australian	70	73	60	76	77	68	60	63	82	81	82	80	75	78	81	81	83	82	82	81	78	79	83	81	69	68	69	68	61	67	69	65	
DED-f	97	97	96	97	96	93	96	97	95	95	93	93	87	95	93	91	85	83	85	80	70	86	85	85	85	82	85	80	60	88	85	65	
diabetes	84	85	84	86	86	77	83	83	81	79	81	76	66	79	81	73	85	85	85	85	83	85	85	85	75	74	74	73	71	70	73	72	
vehicle	95	95	95	95	94	95	95	95	45	42	42	40	22	44	41	28	57	57	56	56	50	49	55	56	46	44	42	45	31	27	42	33	
DED-g	97	97	97	97	96	92	97	97	87	86	86	84	36	85	85	71	95	95	95	94	90	95	95	91	91	91	90	79	92	91	81		
stock	96	96	95	97	97	97	95	95	91	90	86	89	69	95	84	86	71	70	68	66	47	62	68	69	86	86	80	83	68	79	79	78	
vowel	97	97	96	97	95	98	96	96	91	90	90	88	73	89	89	86	82	81	81	77	52	79	81	79	81	79	79	76	41	72	79	61	
credit-g	61	63	55	68	72	64	55	55	64	63	63	64	62	63	63	65	78	77	78	77	74	75	78	76	57	58	57	62	59	59	56	60	
DED-h	91	91	91	92	85	89	90	92	94	94	95	95	83	94	94	90	79	79	79	78	68	78	78	78	66	67	63	77	60	89	61	63	
Ave. OpenML	78	78	74	78	76	74	74	74	73	71	70	69	59	71	69	67	73	72	72	71	65	72	72	72	69	68	67	67	58	68	66	62	
Ave. DED	82	82	79	82	80	80	79	80	85	84	83	82	65	84	82	76	80	78	79	77	67	80	79	79	78	76	77	76	60	79	76	64	
Ave. Overall	79	79	75	79	77	75	75	76	76	74	73	72	61	74	72	69	75	74	74	72	66	74	74	74	71	70	69	69	58	70	69	63	

Appendix C

Domain shift

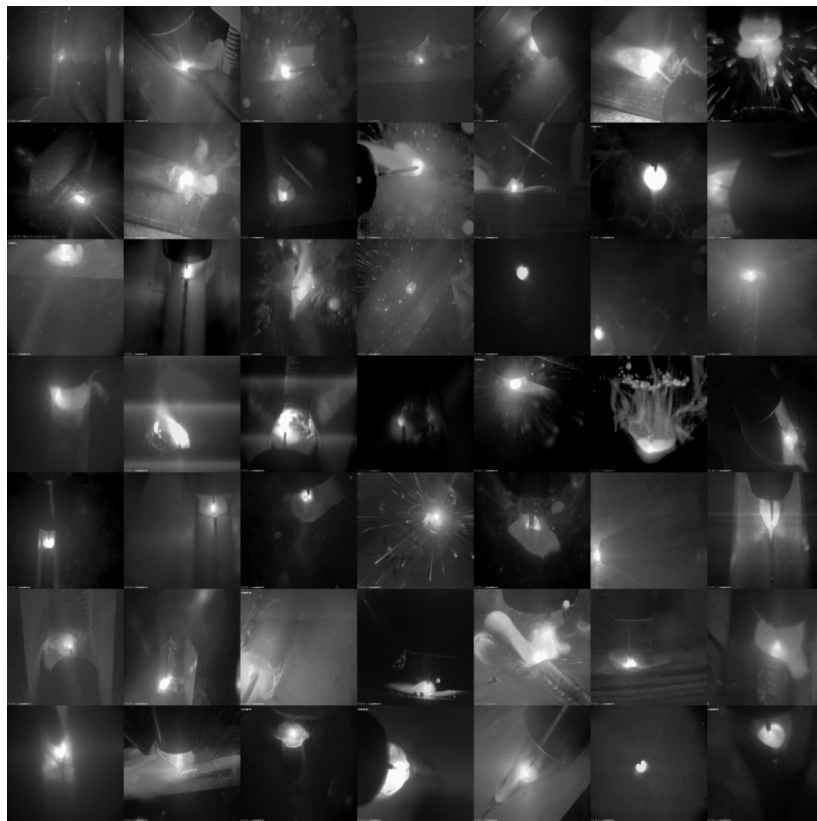


Figure C.1: An example image for each of the 49 different DED processes.

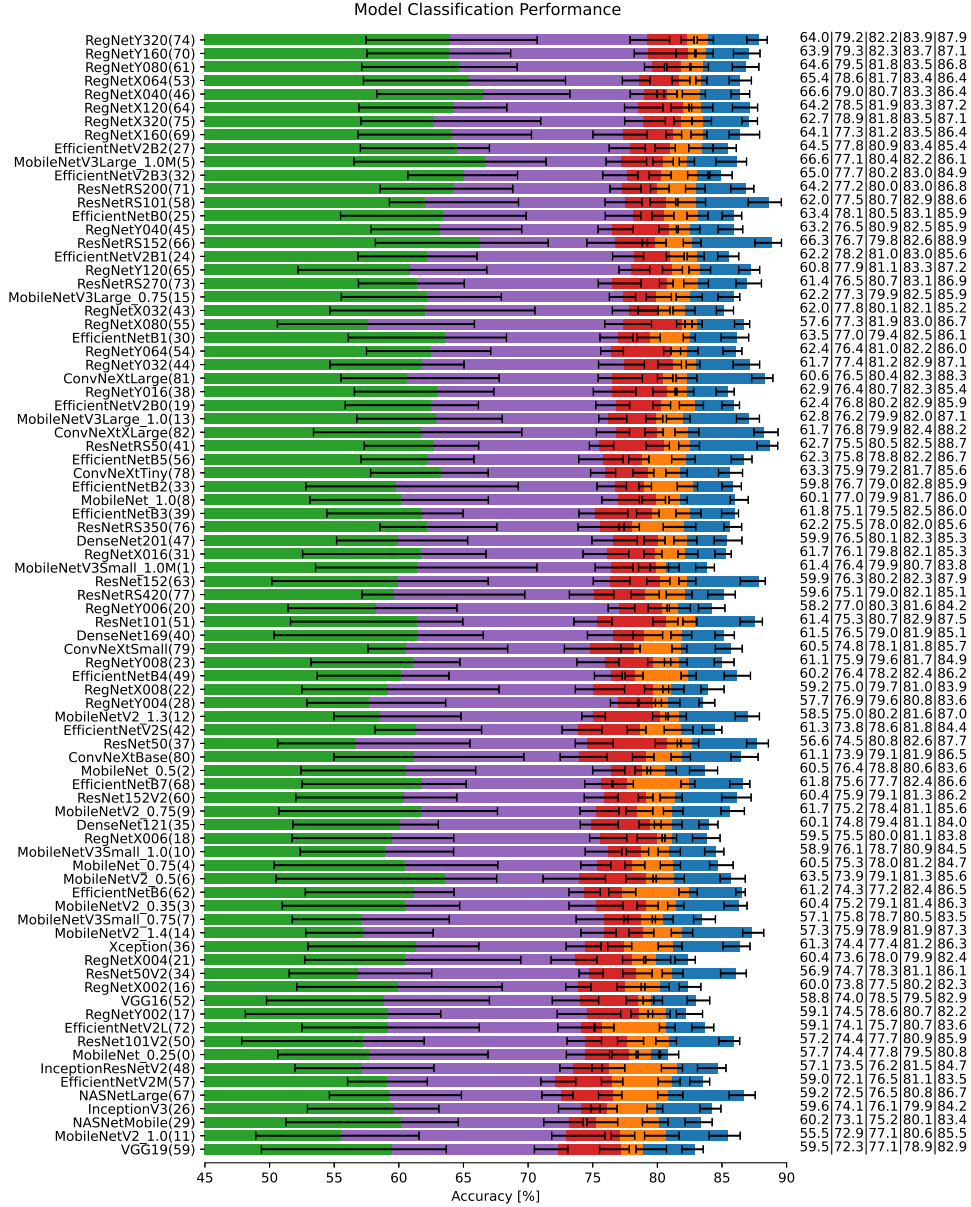


Figure C.2: Classification accuracy of 83 architectures. Colors represent domain shifts. Green (Val D); hardware changes, purple (Val C); parameter changes, red (Val B); dynamics changes, and orange (Val A); uniform random sampling. Blue is training dataset accuracy. Bar widths represent 50% quantiles. Error bars represent 25 and 75% quantiles. Median values are provided on the right.

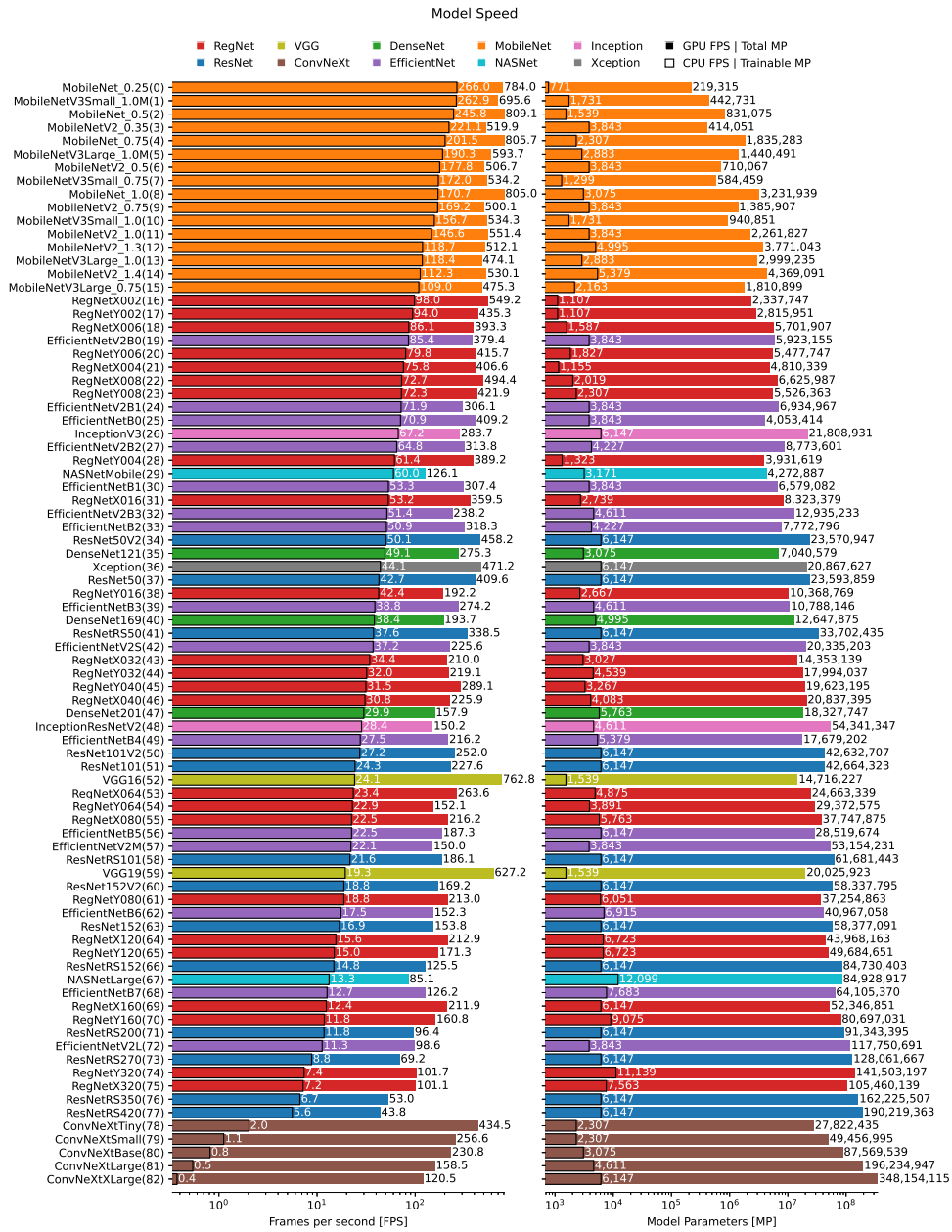


Figure C.3: Inference speed for 83 architectures. Color groups represent a single model CNN architecture family. Bold edged bars represent FPS evaluated on a Intel i7-9700K CPU. bars without edges represent model speed when computation is performed on a Nvidia RTX-3090 GPU using Tensorflow 2.13 with XLA enabled.

Appendix D

AMVOS

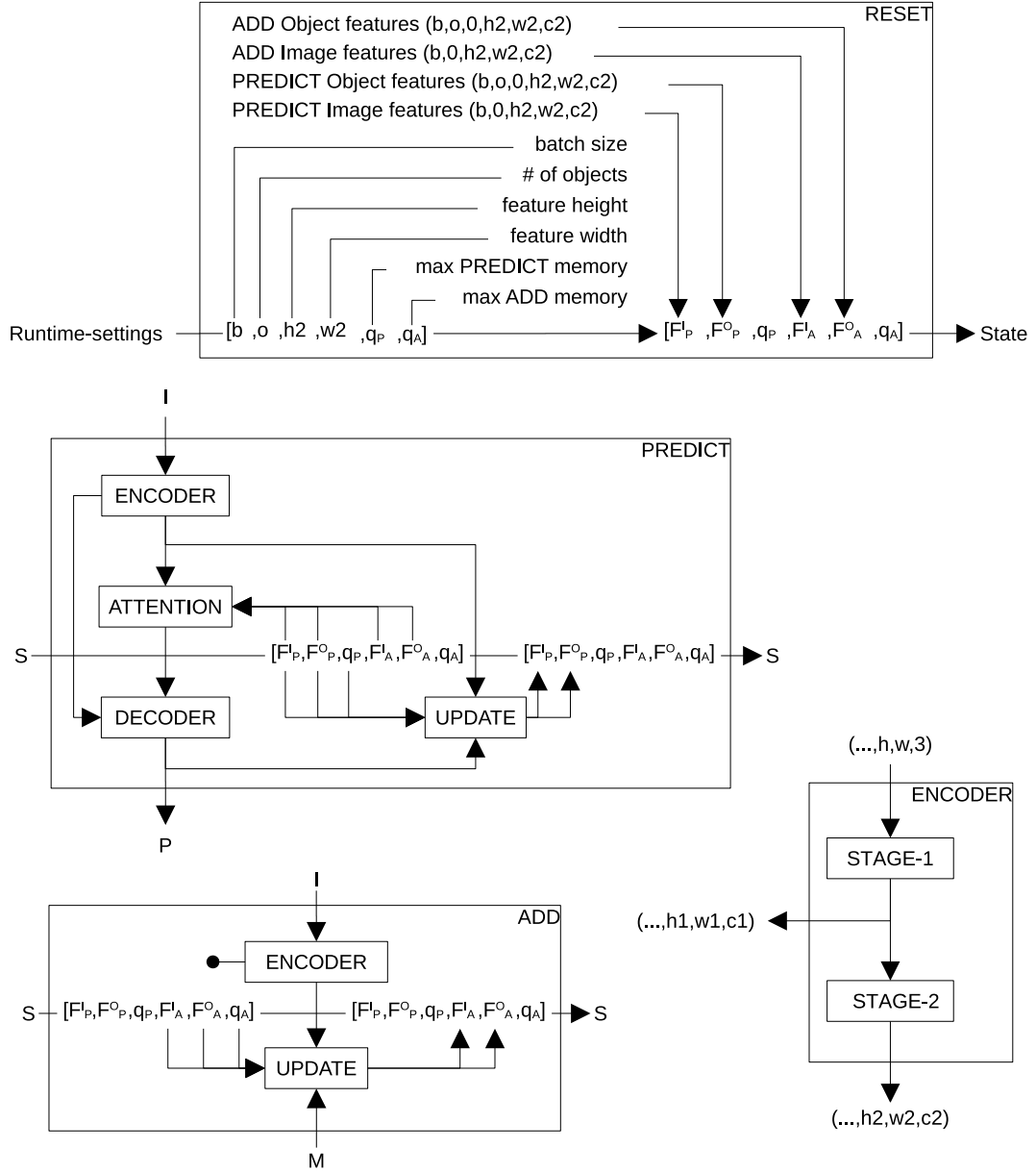


Figure D.1: AMVOS architecture blocks

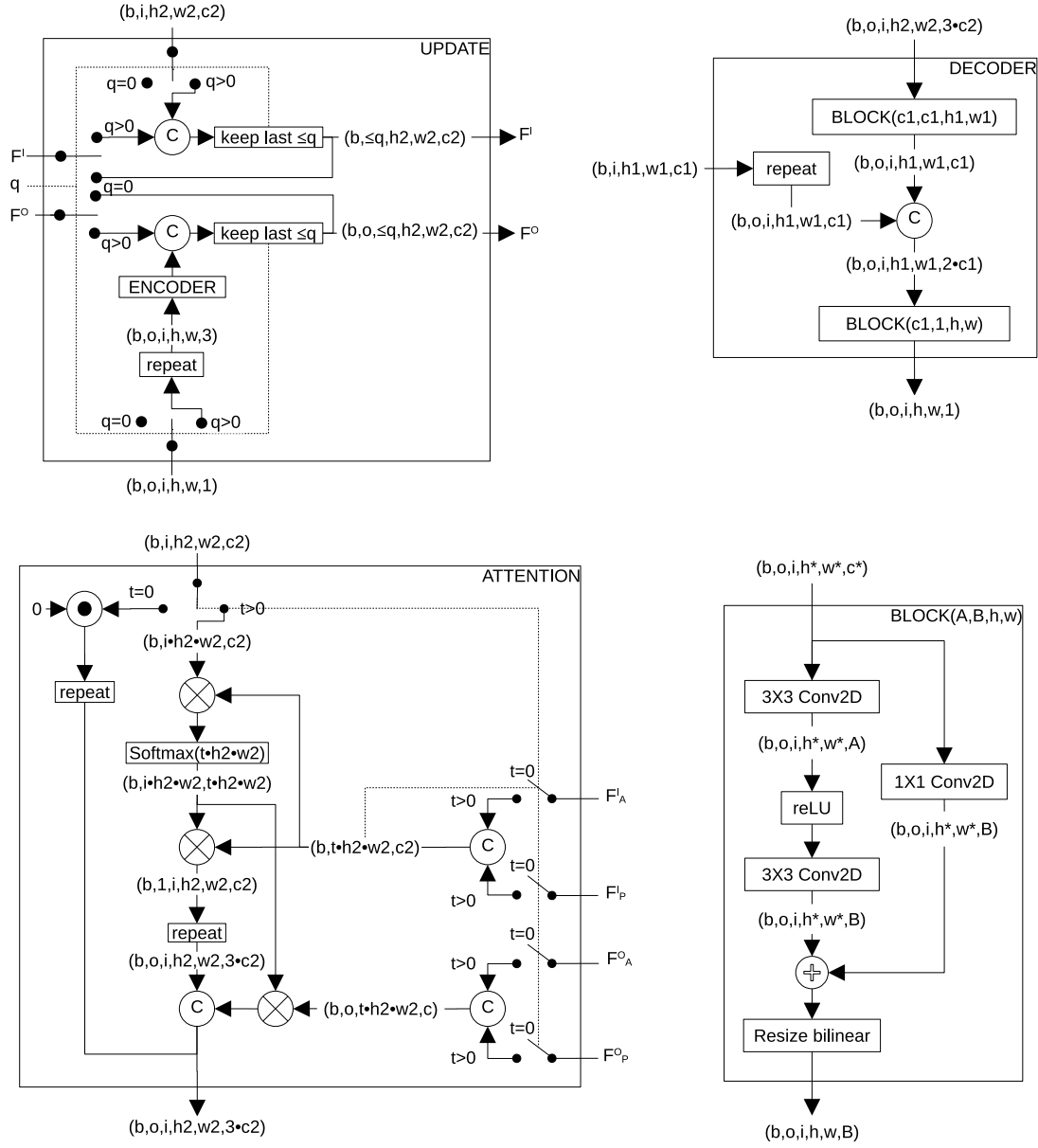


Figure D.2: AMVOS architecture blocks