Leveraging Deep Learning for Energy Consumption Prediction in Selective Laser Sintering

Yixin Li

School of Engineering

Cardiff University

A thesis submitted to Cardiff University

for the degree of Doctor of Philosophy

December 2024

Acknowledgements

My sincere and heavy thanks and appreciation go firstly to my supervisor, Prof. Ying Liu, whose suggestions and encouragement have given me much insight into my research. It has been a great privilege and joy to study under his guidance and supervision, which I will treasure my whole life.

Also, I must express my sincere thanks to Dr. Michael Ryan, who is my co-supervisor, for illuminating guidance and profound knowledge of my research. Working on a very busy schedule, he still gives my other papers a careful reading and detailed comments. I also express my appreciation to Dr. Stephen Robson for his expertise and the critical resources he provides, which are a significant part of my research. Especially this thesis owes a debt to my senior colleagues who have graduated, Dr. Jian Qin, Dr. Chong Chen, Dr. Zheyuan Chen and Dr. Fu Hu, who gave me substantial support and urged me to do better in my studies and research.

At last, I would like to give my heartfelt thanks to my parents, for their endless love and care for me. Whatever I need and wherever I go, they are always there supporting me without any requirements in return. I thank other loving family members, and family is where I can forever turn. I am also extremely grateful to all my friends and other colleagues in my research group who have kindly provided me with assistance and companionship in the course of preparing this thesis.

Abstract

Additive Manufacturing (AM), commonly known as 3D Printing (3DP), is a layer-by-layer manufacturing technique for fabricating objects based on digital models. This technology is widely applied across industries due to its highly customisable and flexible design capabilities. The considerable energy consumption that potentially limits widespread applications is particularly important when seeking suitable and cost-effective manufacturing methods. Energy management and optimisation before or during the process are challenging due to the high demand for data analytics in the dynamic working environment in AM systems. Therefore, Deep Learning (DL) has increasingly been recognised by academia and enterprises to manage energy predictive modelling based on different design-relevant parameters of AM processes. Advanced data analytics and DL techniques are leveraged to develop more accurate and efficient models for predicting and optimising energy consumption in AM systems.

However, traditional energy modelling in AM systems has significant limitations, particularly in handling large and complex datasets in these dynamic environments in AM systems and capturing valuable insights from non-linear relationships between energy consumption and various parameters. This has led to a worldwide recognition of the problems associated with establishing energy predictive models in AM systems. Design for Additive Manufacturing (DfAM) considers energy efficiency with its functionality and manufacturability, and integrating data-driven systems can optimise AM systems. The precise information required to optimise designs could be provided by energy consumption modelling. DfAM helps manufacturers consider energy efficiency at the design stage, leading to more economical and sustainable manufacturing by managing the energy consumption of various design options and optimisation support. DL provides an alternative to building the framework of energy management and optimisation support. Compared to conventional analytical approaches, Deep Neural Networks (DNNs) take significant advantages in handling different data, revealing and predicting complex patterns or insights in AM systems. Differently from other accelerating devices, Field-Programmable Gate Arrays (FPGAs) are often reprogrammable to perform new types of computing tasks. This is due to the

computing capabilities and flexibility, which allow them to work collaboratively with Central Processing Units (CPUs) in terms of training and inference. However, the complexity and memory requirements of predictive models pose challenges, failing to perform edge computing on FPGA platforms directly.

This research is established based on a comprehensive framework for managing and optimising energy consumption and design-relevant parameters, integrating design-relevant parameters with image data to optimise overall energy consumption. The framework is organised into three key topics, which consider a Selective Laser Sintering (SLS) system as the case study. Firstly, a data-driven approach using multi-scale feature fusion techniques is proposed to predict energy consumption from different layer-wise image data, providing insights into the valuable energy consumption patterns. Secondly, to address the challenges of complexities of the model, Knowledge Distillation (KD) is employed, compressing a cumbersome teacher model to a lightweight student model, thereby deploying on the edge device. Finally, Particle Swarm Optimisation (PSO) utilises insights from the lightweight model to optimise design-relevant parameters, providing optimisation support for the case study. The framework offers a potential method to achieve an efficient design with optimal parameter combinations with adjustment of the energy consumption of different prototypes. The framework improves the accuracy of energy consumption predictions, facilitating more energy-efficient AM processes and sustainable manufacturing practices.

Research Achievements

Journal Papers

- **Yixin Li**, Fu Hu, Ying Liu, Michae Ryan and Ray Wang 2023, "A hybrid model compression approach via knowledge distillation for predicting energy consumption in additive manufacturing", International Journal of Production Research

(*Impact factor: 9.2*)

- Fu Hu, Ying Liu, **Yixin Li**, Shuai Ma, Jian Qin, Jun Song, Qixiang Feng, Qian Tang 2023. "Task-driven data fusion for additive manufacturing: framework, approaches, and case studies." Journal of Industrial Information Integration

(*Impact factor: 10.4*)

Conference Papers

- **Yixin Li**, Ying Liu, Stephen Robson, Michael Ryan, "Towards an Energy Consumption Optimisation Framework in Selective Laser Sintering System: Leveraging Deep Learning and FPGA Technologies", presented at 4th ICPR AEM Poznan, Poland, Virtual, 28 June-3July 2024.

- **Yixin Li,** Fu Hu, Michael Ryan, Ray Wang and Ying Liu, "Knowledge distillation for energy consumption prediction in additive manufacturing", presented at 14th IFAC Workshop on Intelligent Manufacturing Systems (IMS 2022), Tel-Aviv, Israel, Virtual,28-30 March 2022.

- **Yixin Li,** Fu Hu, Jian Qin, Michael Ryan, Ray Wang and Ying Liu, "A hybrid machine learning approach for energy consumption prediction in additive manufacturing", presented at 25th International Conference on Pattern Recognition (ICPR 2020), Virtual, 15 January 2021.

- Fu Hu, Jian Qin, **Yixin Li**, Ying Liu and Xianfang Sun, "Deep fusion for energy consumption prediction in additive manufacturing", presented at 54th CIRP Conference on Manufacturing Systems (CMS 2021), Virtual, 22-24 September 2021.

Table of Contents

Acknowledgements ····· **i**

Abstract ····· **ii**

Research Achievements ····· **iv**

Table of Contents ····· **vi**

List of Tables ····· **xiii**

List of Figures ····· **xv**

List of Symbols ····· **xviii**

List of Abbreviation ····· **xxiii**

Chapter 1 Introduction ····· **1**

1.1 Background ····· 1

1.2 Motivations ····· 3

1.3 Research Questions and Objectives ····· 5

1.4 Thesis Outline ····· 7

1.5 Research Contributions ····· 9

Chapter 2 Related Works ····· **11**

2.1 Introduction ····· 11

2.2 An Overview of AM Systems ····· 11

2.2.1 Prevailing Technologies in AM ····· 12

2.2.2 SLS Technology ····· 17

2.2.3 Data Generation and Acquisition in AM Systems ····· 18

2.2.4 Challenges and Quality Considerations in AM ····· 20

- *The Efficiency of the Production Mode* ····· 20
- *Material* ····· 21
- *Costs* ····· 21

2.3 Energy Consumption Analysis and Approaches in AM ····· 22

2.3.1 Early Recognition of Sustainability in AM ····· 22

2.3.2 Initial Energy Consumption Studies ····· 23

2.3.3 Energy Consumption Analysis and ML Integration in AM ····· 24

2.3.4 Advanced Prediction Models and Holistic Energy Optimisation in AM ·· 25

- *DL Applications in AM Energy Consumption Predictive Modelling* ····· 26
- *Predictive Modelling for Energy Optimisation Support in AM* ····· 27

2.4 Advanced Data Analytics in AM ····· 28

2.4.1 Background of DL ····· 29

- *Fully Connected Neural Network* ····· 30
- *Convolutional Neural Network* ····· 31
- *Recurrent Neural Network* ····· 32

2.4.2 Recent Advances in DL in AM ····· 34

- *Material Development* ····· 34
- *Process Optimisation* ····· 35
- *Design Optimisation* ····· 36
- *Quality Control* ····· 37

2.4.3 Challenges and Opportunities in DL for AM ····· 38

2.5 Model Compression Techniques ····· 39

2.5.1 Pruning ····· 39
2.5.2 Quantisation ····· 41
2.5.3 Low-Rank Approximation ····· 43
2.5.4 Knowledge Distillation ····· 43
- *Logit-based Knowledge Distillation* ····· 46
- *Feature-based Knowledge Distillation* ····· 47
- *Relation-based Knowledge Distillation* ····· 49

2.6 Field-Programmable Gate Arrays ····· 51
2.6.1 Fundamentals of FPGAs ····· 51
2.6.2 Accelerating CNNs with FPGAs ····· 53
2.6.3 Smart Manufacturing Applications of FPGAs ····· 54

2.7 Summary ····· 56

Chapter 3 A Framework for Predictive Modelling and Energy Consumption Optimisation in SLS ····· 57

3.1 Introduction ····· 57

3.2 Research Framework for Energy Consumption Prediction in SLS ····· 57
3.2.1 Data Collection and Knowledge Acquisition ····· 60
- ***Data for Multi-scale Feature Fusion*** ····· 61
- ***Data for Optimisation Support*** ····· 62

3.2.2 Image-based Predictive Modelling ····· 65
3.2.3 Model Compression ····· 66
3.2.4 Lightweight Model Deployment ····· 67
3.2.5 Optimisation ····· 68

3.3 Summary ····· 69

Chapter 4 Data-driven Modelling for Energy Consumption Prediction with Knowledge Distillation in SLS ····· 70

4.1 Introduction ····· 70

4.2 Overview of Knowledge Distillation-based Predictive Modelling ····· 70

4.2.1 The Strategy of Ensemble Approach ····· 72

4.2.2 Logit-based KD ····· 75

4.2.3 KD with a Teacher Assistant ····· 77

4.3 Experimental Design and Setups ····· 77

4.3.1 Experiment Setups ····· 78

4.3.2 Evaluation Metrics ····· 81

4.4 Results and Discussions ····· 82

4.4.1 Results of Baseline Models Training ····· 82

4.4.2 Results of KD with Teacher Assistant and Ensemble Teacher Model ····· 84

4.4.3 Results of Energy Consumption Prediction with Distilled Student Models by Using KD ····· 89

4.5 Summary ····· 91

Chapter 5 Leveraging FPGAs and Lightweight Neural Networks for Predictive Modelling in SLS ····· 93

5.1 Introduction ····· 93

5.2 Integration of FPGAs and Lightweight NNs for Predictive Modelling in SLS 93

5.2.1 Multi-scale Feature Fusion for Energy Consumption Modelling ····· 93

5.2.2 Feature-based KD Strategy ····· 95

5.2.3 FPGA-based Prediction Model Implementation ····· 97

- *Student Model for Processing Image-based Data* ····· 97
- *CNN Architecture on the Targeted FPGA Platform* ····· 98

5.3 Experimental Design and Setups ····· 102

5.3.1 Experiment Setups ····· 102

- *Deep Learning-based Energy Consumption Prediction Model* ····· 102
- *Lightweight Model through Feature-based Knowledge Distillation* ···· 103
- *FPGA-accelerated Lightweight Model Development* ····· 103

5.3.2 Evaluation Metrics ····· 104

5.4 Results and Discussions ····· 105

5.4.1 Performance of Teacher Model ····· 105

5.4.2 Results of KD Strategies ····· 107

5.4.3 Performance of Implementation of FPGA-CNN ····· 108

5.5 Summary ····· 110

Chapter 6 FPGA-based Management System for Optimising Energy Consumption in SLS ····· 112

6.1 Introduction ····· 112

6.2 Method of Enhancing Predictions with FPGA-CNN ····· 114

6.2.1 Multi-scale Feature Fusion for Energy Consumption Predictive Modelling ····· 117

- *Enhancing U-Net Performance through Asymmetric Convolution Block* 118
- *Convolution Block Attention Module* ····· 119
- *The Proposed Architecture of U-Net for Energy Consumption Predictive Modelling* ····· 120

6.2.2 Dual KD Strategy ····· 121

6.2.3 Lightweight Energy Consumption Model and FPGA Collaboration ····· 123

6.3 PSO-based Technique for AM Parameters and Energy Optimisation ····· 125

6.3.1 Research Visions and Solutions in FPGA and Predictive Model Integration ····· 125

6.3.2 PSO Technique in the SLS System ····· 127

6.4 Experimental Design and Setups ····· 129

6.4.1 Experiment Setups ····· 131

- *DL-based Energy Consumption Prediction Model* ····· 131
- *Lightweight Model by KD* ····· 132
- *Lightweight Model Acceleration with the Targeted FPGA Platform* ··· 133
- *PSO-based optimisation technique on design-relevant parameters* ···· 134

6.4.2 Evaluation Metrics ····· 135

6.5 Results and Discussions ····· 136

6.5.1 Results of the Multi-scale Feature Fusion Model ····· 136

6.5.2 Results of the Lightweight Model ····· 137

6.5.3 Deployment of the Lightweight Student Model ····· 140

6.5.4 Optimised Energy Consumption with Design-relevant Parameters and Image Features ····· 143

6.6 Summary ····· 151

Chapter 7 Achievement and Conclusions ····· 152

7.1 Achievements ····· 152

7.2 Future Works ····· 155

7.3 Conclusions ····· 156

Bibliography ····· 158

Appendix A Preliminary Study on Energy Consumption Prediction Modelling ····· 187

A.1 Machine Learning ····· 187

- *Support Vector Regression (SVR)* ····· 187
- *Gradient Boost Regression Tree (GBRT)* ····· 188
- *Density-Based Spatial Clustering of Applications with Noise (DBSCAN)* 190
- *Extreme Gradient Boost (XGBoost)* ····· 191

A.2 A Hybrid Machine Learning Approach for Energy Consumption Prediction on Layer-level Data ····· 193

A.3 Particle Swarm Optimisation in Additive Manufacturing ····· 199

Appendix B Datasets Used in This Thesis ····· 201

B.1 Screenshot of Layer-wise Image Data for Training (Partially) ····· 201

B.2 Screenshot of Layer-wise Image Data for Testing (Partially) ····· 203

B.3 The Unit Energy Consumption of Each Layer (Partially) ····· 205

B.4 The Design-relevant Parameters of the Build ····· 205

List of Tables

Table 2.1 Overview of AM technologies. 14

Table 3.1 The descriptions of part-design data. 63

Table 3.2 The descriptions of process-planning data. 64

Table 4.1 Comparative model performance in energy consumption prediction in baseline and ensemble model. 84

Table 4.2 Comparative model performance analysis: baseline, ensemble and teacher assistant models. 85

Table 4.3 Illustration of model performance in terms of distilled student models. ... 86

Table 4.4 Comparative analysis of experimental results based on geometry features and process parameters. 89

Table 5.1 Evaluation metrics. 104

Table 5.2 Comparative analysis of benchmarks and proposed CNN on AM data training performance. 105

Table 5.3 Comparative analysis of benchmarks and proposed CNN based on FLOPs and parameters. 106

Table 5.4 Effectiveness comparison of knowledge distillation variants in terms of plain, logit and feature-based strategies. 107

Table 5.5 Synthesis resource utilisation report. 109

Table 5.6 Power utilisation metrics 109

Table 6.1 The evaluation metrics. 135

Table 6.2 The comparison of baseline and proposed CNN architecture. 137

Table 6.3 The ablation study of the proposed architecture and vanilla U-net with the KD process regarding different KD strategies. 138

Table 6.4 The comparison of SOTA lightweight architecture and distilled student network..... 139
Table 6.5 Optimised CNN module utilisation report. 141
Table 6.6 Detailed report on synthesis resource utilisation of IP cores. 142
Table 6.7 Power utilisation metrics. 142
Table 6.8 The comparison of FPGA and CPU in terms of running performance and power consumption (50MHz period). 143
Table 6.9 Comparison of original and improved build energy consumption (Wh/g) for different algorithms. 144
Table 6.10 Results of design-relevant parameters in Build 1 by using different optimisation algorithms..... 146
Table 6.11 Results of design-relevant parameters in Build 2 by using different optimisation algorithms..... 147
Table 6.12 Results of design-relevant parameters in Build 3 by using different optimisation algorithms..... 148

List of Figures

Figure 1.1 Key milestones in AM process development. 1

Figure 2.1 Mainstream AM technology categories. 13

Figure 2.2 SLS system working principle. 18

Figure 2.3 The relationship between AI, ML and DL. 29

Figure 2.4 FCNN architecture overview. 31

Figure 2.5 CNN architecture overview. 32

Figure 2.6 RNN architecture overview. 33

Figure 2.7 Pruning technique. 41

Figure 2.8 Quantisation technique. 42

Figure 2.9 Knowledge distillation in teacher-student architecture. 44

Figure 2.10 Logit (response)-based KD architecture. 47

Figure 2.11 Feature-based KD architecture. 49

Figure 2.12 Relation-based KD architecture. 50

Figure 2.13 Detailed FPGA architecture including programmable logic and I/O. 52

Figure 3.1 Framework for predictive modelling and energy consumption optimisation in SLS. 59

Figure 3.2 Categorisation of data collected in the SLS process. 60

Figure 3.3 The sliced image data of a sample and the distribution of unit energy consumption (Wh/g) of one product. 61

Figure 3.4 The samples of build for optimisation. 63

Figure 4.1 Workflow diagram of the proposed methodology. 72

Figure 4.2 Ensemble method in ML. 74

Figure 4.3 Workflow of model training and implementation for the proposed approach. 79

Figure 4.4 RMSE comparison of trained models in terms of CNNs and ensemble predictions. 83
Figure 4.5 MAE comparison of trained models in terms of CNNs and ensemble predictions. 83
Figure 4.6 Comparative RMSE analysis of distilled student model A and B with Original model. 87
Figure 4.7 Comparative MAE analysis of distilled student model A and B with Original model. 88
Figure 4.8 Unit energy consumption prediction accuracy for model A. 90
Figure 4.9 Unit energy consumption prediction accuracy for model B. 90
Figure 5.1 Detailed workflow of FPGA-CNN integration for predictive energy modelling. 94
Figure 5.2 Student model architecture overview. 98
Figure 5.3 Block diagram of a CNN designed for FPGA deployment and acceleration. 99
Figure 5.4 Fundamentals of convolutional operation within CNNs. 100
Figure 5.5 Structural architecture of a convolutional block in CNNs. 100
Figure 5.6 Detailed structure of a 5×5 convolution operation including multiplications and accumulations. 101
Figure 5.7 Maxpooling layer implementation in CNNs. 102
Figure 5.8 PYNQ-Z2 as the targeted platform. 104
Figure 5.9 Comparative predictive performance of teacher, distilled student and original student models. 108
Figure 6.1 Overview of energy consumption modelling in AM monitoring with design parameters, image features, and energy-relevant data. 116

Figure 6.2 Detailed workflow of FPGA-based predictive modelling for SLS energy consumption. 117
Figure 6.3 Teacher model architecture for multi-scale feature fusion. 121
Figure 6.4 Student model architecture overview. 124
Figure 6.5 Integrated solution of FPGA-based CNNs for AM energy consumption analysis. 126
Figure 6.6 Integration of PSO in the energy consumption prediction model. 127
Figure 6.7 Experimental setup for SLS energy optimisation monitoring. 130
Figure 6.8 Xilinx PYNQ-Z2 platform for data processing in the experiment. 133
Figure 6.9 The overlay of the project for the experiment. 134
Figure 6.10 The actual and predicted energy consumption of samples. 140
Figure A. 1 Divided hyperplanes to separate the two types of training samples. 188
Figure A. 2 The Multi-Source Data Collection from AM System. 194
Figure A. 3 The Framework of Proposed Methodology for Energy Consumption Prediction. 195
Figure A. 4 Comparison of RMSE of XGBoost and benchmarks. 196
Figure A. 5 Comparison of MCC of XGBoost and benchmarks. 197
Figure A. 6 The prediction result between predicted values and original values..... 197

List of Symbols

$\boldsymbol{\alpha}$: Hyperparameter used to balance the soft label loss.

$\boldsymbol{\alpha_i}$ and $\boldsymbol{\alpha_i'}$: Lagrange multipliers

$\boldsymbol{\beta}$: Hyperparameter used to balance the hard label loss, where $\boldsymbol{\beta = (1 - \alpha)}$

$\boldsymbol{\gamma}$: Hyperparameter for L2 regularisation

$\boldsymbol{\gamma}$: Learning rate for each regression tree

$\boldsymbol{\kappa(x, x_i)}$: Kernel function

$\boldsymbol{\lambda}$: Hyperparameter for L2 regularisation

$\boldsymbol{\Omega(f_k)}$: Regularisation term for a single tree

$\boldsymbol{\phi(x)}$: Function used in Smooth L1 loss

$\boldsymbol{\phi(x_i)}$: Mapping function for kernel trick

$\boldsymbol{\Phi_t}$: Transformation function for the teacher network's features

$\boldsymbol{\Phi_s}$: Transformation function for the student network's features

$\boldsymbol{\overline{a}}$: Mean value of actual value

$\boldsymbol{a_t}$: Actual value

$\boldsymbol{b}$: Bias

$\boldsymbol{C}$: Total number of input channels

$\boldsymbol{c_1}$: Acceleration coefficient

$\boldsymbol{c_2}$: Acceleration coefficient

$\boldsymbol{D}$: Image-based feature dataset of the selected samples

$\boldsymbol{dist(p, q)}$: Distance function, often Euclidean distance

$\boldsymbol{E}$: Predicted energy consumption

$\boldsymbol{E_T}$: Total energy usage

$\boldsymbol{E_U}$: Specific (Unit) energy consumption

$\boldsymbol{F}$: Convolutional kernel

$\boldsymbol{F^{(j)}_{:,:,k}}$: k-th input channel of the j-th filter

$\boldsymbol{F_m(x)}$: Model after m iterations

$\boldsymbol{F'}$: Feature map after applying channel attention

$\boldsymbol{F''}$: Feature map after applying spatial attention

$\boldsymbol{f(X)}$: Objective function

$\boldsymbol{f_k}$: k-th tree model

$\boldsymbol{f_s(x)}$: Intermediate features of the student network

$\boldsymbol{f_t(x)}$: Intermediate features of the teacher network

$\boldsymbol{G_{best}}$: Global best position in the search space

$\boldsymbol{h(x)}$: Soft label derived from the outputs of multiple teacher networks

$\boldsymbol{h(x)}$: weak learner, a regression tree

$\boldsymbol{L}$: Distillation loss function

$\boldsymbol{L(y_i, \gamma)}$: Loss function to be minimised

$\boldsymbol{l(y_i, \hat{y}_i)}$: Loss function for individual instance predictions

$\boldsymbol{L_{distil}}$: Distillation loss

$\boldsymbol{L_{FeaD}}$: Distillation loss based on intermediate features

$\boldsymbol{L_{feature}}$: Loss from the feature layer

$\boldsymbol{L_F}$: Similarity function comparing feature maps of teacher and student networks

$\boldsymbol{L_{hard}}$: Soft label loss derived from the cross-entropy between the original label and the student output

$\boldsymbol{L_{soft}}$: Soft label loss derived from the cross-entropy between the soft targets and the student output

$\boldsymbol{L_{student}}$: student total loss function, the sum of $\boldsymbol{L_{soft}}$ and $\boldsymbol{L_{hard}}$

$\boldsymbol{L_{SmoothL1}}$: Smooth L1 loss function

$\boldsymbol{L_{task}}$: Task-specific loss, representing Smooth L1 loss

$\boldsymbol{L_{total}}$: Total loss function, sum of L_{task} and L_{distil}

$\boldsymbol{L_{label}}$: Loss from the actual label

$\boldsymbol{M}$: Total number of features

$\boldsymbol{M_T}$: Mass of the total part

$\boldsymbol{M_c(F)}$: Channel attention map

$\boldsymbol{M_s(F')}$: Spatial attention map

$\boldsymbol{M_{:,:,k}}$: Input channel feature map at the k-th filter

$\boldsymbol{MinPts}$: Minimum number of points required for a dense region

$\boldsymbol{N}$: Total number of classes or instances in the dataset

$\boldsymbol{N_{Eps}}$: Neighbourhood of radius Eps

$\boldsymbol{n}$: Number of data points

$\boldsymbol{O_{:,:,j}}$: Output channel feature map at the j-th filter

$\boldsymbol{P_{best,id}}$: Best position of particle i in dimension d

$\boldsymbol{p_t}$: Predicted value

$\boldsymbol{\overline{p}}$: Mean value of predicted value

$\boldsymbol{Q}$: Quantised value

$\boldsymbol{q_i}$: Soft target distribution for class i from the teacher model

$\boldsymbol{r_1}$: Random numbers ranging from 0 to 1

$\boldsymbol{r_2}$: Random numbers ranging from 0 to 1

$\boldsymbol{S}$: Optimised student model

$\boldsymbol{S}$: Scaling factor

$\boldsymbol{S_A}$: Variance of actual data

$\boldsymbol{S_P}$: Variance of predicted data

$\boldsymbol{S_{PA}}$: Covariance between predicted and actual data

$\boldsymbol{T}$: Number of nodes in a tree model

$\boldsymbol{T}$: Hyperparameter temperature used in the softmax function to smooth probability distribution

$\boldsymbol{T}$: Total number of models

$\boldsymbol{v}$: Original floating-point value

$\boldsymbol{v_i(t+1)}$: Velocity of particle i at time t

$\boldsymbol{v_{id}}$: Velocity of the particle in a D-dimensional space

$\boldsymbol{w}$: Inertia weight

$\boldsymbol{w}$: Collection of scores at the leaves

$\boldsymbol{w}$: Weight factor

$\boldsymbol{w_{ij}}$: Weights assigned to the soft labels q_{ij} from the teacher model

$\boldsymbol{w_i}$: Soft label derived from the outputs of multiple teacher networks

$\boldsymbol{X}$: Vector of design parameters

$\boldsymbol{x_1, x_2, x_3, \ldots x_n}$: Different design-relevant parameters

$\boldsymbol{x_i(t)}$: Position of particle i at time t

$\boldsymbol{x_i(t+1)}$: Position of particle i at time $t+1$

$\boldsymbol{y}$: Ground truth labels

$\boldsymbol{y_i}$: Actual value for the i-th instance

$\boldsymbol{\hat{y}_i}$: Predicted value for the i-th instance

$\boldsymbol{Z}$: Zero point

$\boldsymbol{z_i}$: output of the teacher model for class i

List of Abbreviation

AC: Asymmetric Convolution

AM: Additive Manufacturing

ANN: Artificial Neural Network

ASIC: Application-Specific Integrated Circuit

BJT: Binder Jetting

BRAM: Block Random Access Memory

CAD: Computer-Aided Design

CBAM: Convolutional Block Attention Module

CLB: Configuration Logic Block

CNN: Convolutional Neural Network

CPU: Central Processing Unit

DBSCAN: Density-Based Spatial Clustering of Applications with Noise

DED: Direct Energy Deposition

DfAM: Design for Additive Manufacturing

DL: Deep Learning

DNN: Deep Neural Network

DT: Decision Tree

EBM: Electron Beam Melting

FCNN: Fully Connected Neural Network

FDM: Fused Deposition Modelling

FF: Flip-Flops

FPGA: Field-Programmable Gate Array

FPN: Feature Pyramid Network

FSM: Finite State Machine

GBRT: Gradient Boosting Regression Tree

GBDT: Gradient Boosting Decision Tree

GPU: Graphical Processing Unit

GRU: Gated Recurrent Unit

IC: Integrated Circuit

I/O: Inputs /Outputs

I4.0: Industry 4.0

IoT: Internet of Thing

KD: Knowledge Distillation

KL: Kullback-Leibler

LCA: Life Cycle Assessment

LOM: Laminated Object Manufacturing

LS: Laser Sintering

LUT: Look-Up Table

LSTM: Long-Short-term Memory

MAE: Mean Absolute Error

MAC: Multiply and Accumulate

MCC: Model Correlation Coefficient

MEX: Material Extrusion

ML: Machine Learning

MLP: Multilayer Perceptron

MJT: Material Jetting

MSE: Mean Square Error

NN: Neural Network

PBF: Powder Bed Fusion

PL: Programmable Logic

PS: Processing System

PSO: Particle Swarm Optimisation

PYNQ: Python Productivity for Zynq

RAM: Random-Access Memory

ReLU: Rectified Linear Unit

RGB: Red, Green and Blue

RNN: Recurrent Neural Network

RMSE: Root Mean Square Error

ROM: Read-Only Memory

SEC: Specific Energy Consumption

SHL: Sheet Lamination

SLA: Stereolithography Apparatus

SLM: Selective Laser Melting

SLS: Selective Laser Sintering

SPP: Spatial Pyramid Pooling

SVR: Support Vector Regression

SVM: Support Vector Machine

SOTA: State-of-the-Art

VPP: Vat Photopolymerisation

WAAM: Wire Arc Additive Manufacturing

XGBoost: Extreme Gradient Boost

3DP: 3D Printing

Chapter 1 Introduction

1.1 Background

Additive Manufacturing (AM), commonly known as 3D Printing (3DP), fabricates parts with complex geometries layer-by-layer directly from Computer-Aided Design (CAD) models (ISO/ASTM 2013). Figure 1.1 demonstrates a significant milestone in the development of prevailing AM technologies and systems. Early AM equipment using Laser Sintering (LS) and photopolymerisation technology emerged in the 1980s (Jiménez et al. 2019) and became popular in the early 2010s because of the emergence of low-cost and desktop 3D printers (Sotoodeh 2022). In the following decades, the emergence of AM has attracted the attention of academia and manufacturers due to its design freedom and flexibility. AM system embraces the development of modern machinery, material science, and software development, and is now accessible across various industries. In the context of Industry 4.0 (I4.0) framework, AM integrates technological and industrial development, such as the Internet of Things (IoT), big data and physical entities, including sensors (Prashar et al. 2023). The AM system could be applied in a range of commercial applications without expensive tooling including aerospace, medical equipment, and prototyping due to its high customisation and design freedom (Pérez et al. 2020). This technique would reduce design time, enhance and improve product quality, and lower the costs associated with repeated manufacturing (Horn and Harrysson 2012).

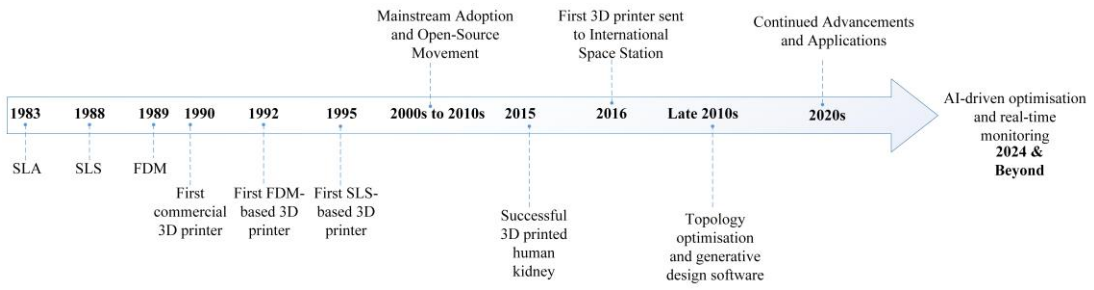


Figure 1.1 Key milestones in AM process development.

A typical AM system consists of six procedures: (1) converting the 3D model to STL file format, (2) determining part orientation, (3) adding support structures, (4) generating sliced files for layer-by-layer fabrication, (5) fabrication, and (6) post-processing to finish the printed objects (Wang and Alexander 2016). Compared to conventional manufacturing processes that rely on tooling and machining, AM has the potential to fabricate complex structural components that were previously constrained by conventional manufacturing (Watson and Taminger 2018). Furthermore, AM has shown its merits of customisation and relatively lower cost when manufacturing small batches of builds (Kellens et al. 2017). The diversity of AM technologies contributes to the selection of the most suitable approach according to parameter requirements and design specifications. Each AM technologies have unique characteristics and working principles, as shown below:

- **Vat Photopolymerisation (VPP)** techniques utilise lasers to cure liquid photopolymer resin into solid objects (Zhang et al. 2021).
- **Material Jetting (MJT)** techniques selectively deposit feedstock material (Gülcan et al. 2021).
- **Binder Jetting (BJT)** techniques selectively deposit a liquid bonding agent onto a powder bed to adhere to the material (Lores et al. 2019).
- **Material Extrusion (MEX)** processes extrude materials from the nozzle layer upon layer in the process (Braconnier et al. 2020).
- **Power Bed Fusion (PBF)** techniques utilise concentrated energy beams to fuse the region of the powder bed (Dev Singh et al. 2021).
- **Sheet Lamination (SHL)** manufacturing processes involve bonding sheets of materials to fabricate the final part (Lores et al. 2019).
- **Direct Energy Deposition (DED)** utilise thermal energy to melt and fuse deposited materials (Tang et al. 2020).

Due to the different working principles and materials used, modelling energy consumption patterns and relative optimisation support raises a significant concern in different AM technologies, which requires a more focused and smart solution in the production life cycle (Majeed et al. 2021). Energy optimisation and management has been a high active research area in the AM system with challenges of developing an energy consumption model (Yang et al. 2017), which could improve the long-term viability and production efficiency of a product lifecycle (Dunaway et al. 2017). Nowadays, researchers and manufacturers have increasingly focused on this aspect. AM has the potential to achieve higher production yields, resulting in an increasing amount of energy consumption (Freitas et al. 2016). Early incorporation of eco-design or Life Cycle Assessment (LCA) in the design and manufacturing process is essential to assist designers and operators in energy management, decision-making, and process optimisation (Kellens et al. 2017).

1.2 Motivations

Selective Laser Sintering (SLS) is a 3D printing technique that uses a high-power laser to sinter powdered material into solid objects, layer by layer, based on CAD models, which is widely used in aerospace, medical fields and prototypes at the early stage (Kellens et al. 2011). SLS is a typical AM technology in which the energy is used to power high-intensity lasers, heating systems, and complex mechanical movements (Yehia et al. 2024). SLS machines consume significant energy, which is a major barrier to their widespread adoption and large-scale production (Hu et al. 2023). Advanced data analytics for managing energy consumption is crucial for AM systems, which can uncover insights into energy usage (Liu et al. 2018b). Data are collected from different sources such as the working environment, design specifications, process parameters and materials (Qin et al. 2018). In such a complex and dynamic scenario in AM systems, traditional energy modelling approaches are inadequate for capturing and connecting the complex relationships between various impact factors (e.g., layer-wise images and design-relevant parameters) and energy consumption. Existing studies regarding SLS systems have focused on impact parameters related to energy consumption prediction (Baumers et al. 2011; Paul and Anand 2012; Baumers et al.

2013; Liu et al. 2018b; Peng et al. 2018; Watson and Taminger 2018), while part geometry of layer-wise images of printed prototypes has not been sufficiently considered as a critical impact factor to energy consumption. There are three motivations below:

Advanced models for complex systems

DL has the ability to deal with complex nonlinear relationships between impact factors, improving the accuracy of energy consumption modelling (Qin et al. 2022). DL is a data-driven approach that requires substantial data with higher dimensions to train models (LeCun 2019), while there will be a large amount of data generated during the SLS production process, offering a sufficient basis for the training of DL models. In the case study, layer-wise images from the sliced CAD model are utilised as the input, and the image features would have the potential to affect the level of energy usage. To develop an image-based model, more advanced data-driven modelling is required by using multi-scale feature fusion approaches. By analysing and extracting image data, advanced analytics can uncover valuable insights from the correlation between specific layer-wise images, design-relevant information, and energy consumption. The energy prediction model is employed, with multi-scale feature fusion to focus on part design details, thoroughly considering detailed patterns and geometrical information for a more precise analysis of energy consumption.

Lightweight models for FPGA deployment

Energy consumption patterns may vary over time, and SLS machines could be equipped with real-time data acquisition and prediction platforms. This platform can be performed and deployed on the edge, providing real-time data input to the energy prediction models thereby enabling dynamic adjustment and optimisation support for energy consumption and design-relevant parameters. Due to parallel processing capability and low power consumption, edge computing platforms such as FPGAs play a vital role in accelerating model inference (Biookaghazadeh et al. 2018). Besides, FPGAs can be customised based on specific tasks (i.e., energy consumption prediction and feature fusion) in SLS systems due to their hardware programmability. However,

the limited resources on FPGAs prevent deep models from direct deployment and acceleration. To enable efficient predictive modelling on the FPGA platform, lightweight models are developed using model compression techniques such as Knowledge Distillation (KD) (Hinton et al. 2015). This technique allows lightweight networks to learn from cumbersome networks, which manage the complexity of AM data with minimal computational resources (Wang et al. 2024a).

Bridging design parameters and energy efficiency

Particle Swarm Optimisation (PSO) is crucial for optimising a series of design-relevant parameters and energy consumption in AM scenarios. In the case study, this optimisation approach will identify design-relevant parameters corresponding to optimal energy consumption. The lightweight model can process image data and provide features and predicted energy, which are integrated with design-relevant parameters. Subsequently, an optimisation algorithm will be employed to minimise the build energy consumption based on the combination of the optimal design-relevant parameters. This framework will provide valuable insights for adjusting AM processes by managing and analysing energy consumption to enhance overall sustainability in AM processes.

1.3 Research Questions and Objectives

The research questions are demonstrated in the following:

1. ***What lightweight Deep Learning architectures and model compression techniques can be developed to effectively analyse layer-wise image data for energy consumption prediction in SLS when deployed on FPGA platforms?***

2. ***How can the inherent parallel processing and reconfigurability of FPGAs be exploited to enhance the performance and energy efficiency of lightweight neural networks for predictive modelling in AM?***

3. ***What are the essential steps and design considerations for integrating an FPGA-based monitoring system for real-time energy consumption analysis in AM, and how does this system enable dynamic optimisation support of energy usage?***

To address these research questions, the following objectives will be conducted:

1. To develop and validate a multi-scale feature fusion model to improve the accuracy and efficiency of a specific AM process using layer-wise image data derived from CAD models. Under the teacher-student architecture by using Knowledge Distillation (KD) strategies, this model serves as the teacher model which will capture and integrate features at different scales, focusing on image features that could affect the energy consumption of each layer.

2. To investigate and implement the KD strategy to develop a lightweight energy prediction model replacing the feature fusion model to predict energy consumption in an SLS system. This lightweight model is regarded as the student model. This lightweight model will maintain high predictive performance while reducing computational resources, making it suitable for deployment on resource-constrained devices.

3. To deploy the lightweight model on FPGAs to accelerate image data processing of the student model and provide the features and predictions of energy. Before

deployment, the parameters of the lightweight model are further quantised to accommodate the resources available on the FPGA. This deployment will facilitate faster feature extraction. This aims to achieve a nearly real-time environment when predicting energy consumption on the edge device.

4. To leverage the PC-FPGA collaboration, the PC is equipped with the targeted FPGA platform for accelerating the inference of the lightweight model. After that, the optimisation algorithm determines the optimal design-relevant parameters, followed by minimising energy consumption in builds. This will help in decision-making during the design and manufacturing phases by offering recommendations and optimisation support.

Details of these objectives and research are demonstrated in Chapters 3, 4, 5, and 6.

1.4 Thesis Outline

The thesis is organised into seven chapters and the outline is listed below:

Chapter 1 introduces an overview of the research background, motivations, research questions, objectives, and contributions of the research work.

Chapter 2 reviews the current studies on different AM processes, energy consumption analytics and advanced data analytics techniques. The chapter summarises AM and discusses the seven prevailing processes with a particular focus on Selective Laser Sintering (SLS). The chapter subsequently examines the types of data used in AM, current constraints and energy consumption analytics. Furthermore, the chapter introduces DL-based and advanced data-driven approaches, followed by a focus on

Knowledge Distillation (KD) techniques. Finally, this chapter reviews the background and application of FPGAs in smart manufacturing.

Chapter 3 presents a predictive modelling framework aimed at optimising energy consumption in an SLS system, incorporating multi-scale feature fusion, model compression, lightweight model deployment, parameter optimisation and decision support.

Chapter 4 describes the preliminary contributions offering the theoretical foundations of the whole research, which involves developing a predictive model for predicting energy consumption based on layer-wise images. In detail, this part of the research focuses on an ensemble model serving as the teacher model under the scheme of teacher-student architecture, followed by exploiting a teacher-assistant model at the intermediate position in conventional teacher-student architecture to mitigate the gap in the learning capacity of the student model. In addition, this chapter provides primary insights into feature extraction for images by using the ensemble technique and the KD technique for lightweight modelling.

Chapter 5 targets to mitigate the limitations of ensemble learning on layer-wise images by employing a multi-scale feature fusion model as the teacher model. By using KD, the student model is deployed on the targeted FPGA platform, where the architecture of the student model must be redesigned to accommodate the resource availability on the targeted FPGA platform. This chapter highlights the multi-scale feature fusion technique for image data and employs a different KD strategy to train the student model. Besides, the FPGA-based implementation is validated to identify the effectiveness of the purposed methodology.

Chapter 6 extends and integrates the previous research outcomes to a more comprehensive data-driven approach utilising multi-scale feature fusion and acceleration with the targeted FPGA platform. The output features and predictions can support design-relevant parameters for optimisation. By using Particle Swarm Optimisation (PSO), the optimised combinations of those parameters help in minimising the energy consumption of the selected build. This chapter focuses on the optimisation of different parameter combinations and the minimal energy consumption of the build, providing guidance and decision support for part designers and process operators.

Chapter 7 concludes the thesis and presents the achievements of the research. It also discusses the limitations of the research and suggests directions for future work.

1.5 Research Contributions

This thesis makes several significant contributions to the broader body of knowledge, particularly in the development of advanced energy prediction models, and the framework of energy consumption predictions and optimisation support in SLS. Each contribution corresponds to the subsequent chapters 3 to 6.

1. The research developed a comprehensive framework for energy management and optimisation support in a targeted SLS system. The technical framework is illustrated for energy predictive modelling at the I4.0 level by leveraging an advanced data-driven approach to integrate insights from image-based data, energy-relevant data and design-relevant data.

2. An accurate energy consumption prediction contributes to energy management in SLS. At the beginning of the framework, the research involves developing a data-

driven approach to integrating layer-wise images derived from CAD models and unit energy consumption measured by a power meter. The multi-scale feature fusion model, U-Net, is utilised with an attention mechanism to achieve an accurate prediction of energy usage. This is the preliminary stage before deployment and this model serves as the teacher model in the KD process.

3. KD strategy allows for the development of a lightweight model based on the knowledge of a complex model rather than designing a new network architecture. The teacher model is associated with a multi-scale feature fusion model, while the student model is the lightweight model, thus maintaining effectiveness and performance. To bridge the performance gap between different models, KD techniques such as logit-based, feature-based and dual strategy are employed. By leveraging KD, a lightweight model is obtained and subsequently deployed and accelerated on resource-limited FPGA platforms. Such implementation and PC-FPGA collaboration contribute to edge computing for nearly real-time inference on the lightweight energy prediction model.

4. The proposed framework leverages the collaboration of the data-driven approach and the FPGA to process different data and analyse energy consumption. Particle Swarm Optimisation (PSO) is employed to optimise the combination of design-relevant parameters, minimising the energy consumption based on the predicted energy consumption and features obtained from the FPGA and lightweight student model. Optimised parameter combinations and minimised energy consumption provide recommendations for part designers and process operators. The framework supports informed decision-making and enhances the energy management and operational efficiency of the targeted AM system. Furthermore, this framework could be applied to monitor the in-situ behaviour of printed objects in the future.

Chapter 2 Related Works

2.1 Introduction

This chapter provides an overview of Additive Manufacturing (AM) technologies, including seven promising technologies in Section 2.2. In Section 2.3, the overview of Selective Laser Sintering (SLS) as one of the Powder Bed Fusion (PBF) technologies will be emphasised, as it is used as the case study throughout the entire research route. This will be followed by a discussion on the types of data within the AM system and the current constraints in AM. In Section 2.3, energy consumption and sustainability in different AM systems will be discussed, including reviews of different analytical approaches to energy consumption, Machine Learning (ML)-based prediction models and predictive modelling for optimisation in AM systems. Section 2.4 will discuss Deep Learning (DL) and advanced data analytics in AM, including current DL algorithms and recent advances in AM. In Section 2.5, model compression techniques, especially the Knowledge Distillation (KD) technique including its background and different distillation strategies will be described. In Section 2.6, a background of FPGAs and their applications in smart manufacturing will be demonstrated.

2.2 An Overview of AM Systems

AM, as a promising manufacturing practice, has contributed to the advancement of the Industry 4.0 (I4.0) environment (Haleem and Javaid 2019). An increasing diversity of customer demands results in complex situations when developing products in a customised manner. This trend has led to the collection and analysis of data to make intelligent decisions regarding automation (Ahuett-Garza and Kurfess 2018). In I4.0, physical entities such as machines and workpieces are combined in embedded systems that collect data, followed by the connection with the network (Thoben et al. 2017).

In this section, seven prevailing AM technologies are introduced in Section 2.2.1. Section 2.2.2 describes the details of SLS. Subsequently, the data generation and acquisition process are described in Section 2.2.3. Finally, the current constraints and quality considerations of AM systems will be discussed.

2.2.1 Prevailing Technologies in AM

According to Figure 2.1, AM falls into three main types based on the physical state of the material used during the manufacturing process including liquid-based systems, solid-based systems, and powder-based systems. In more detail, this can be further categorised into seven mainstream AM technologies: Vat Photopolymerisation (VPP), Material Jetting (MJ), Binder Jetting (BJ), Material Extrusion (MEX), Powder Bed Fusion (PBF), Sheet Lamination (SHL) and Directed Energy Deposition (DED). Table 2.1 summarises an overview of AM technologies and their characteristics in terms of categories, working principles, processes, material used, advantages and disadvantages.

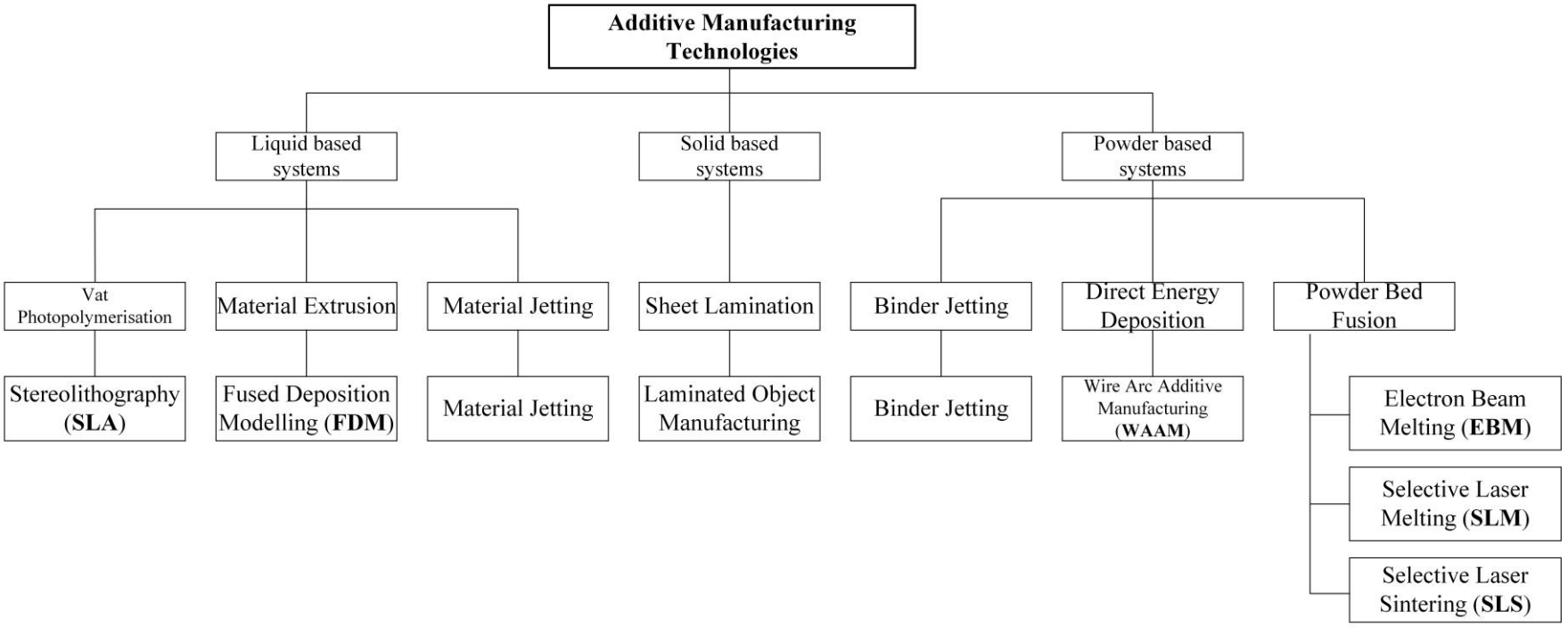
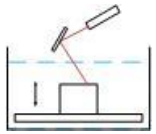
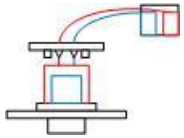
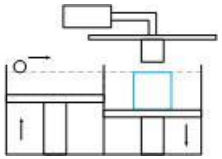
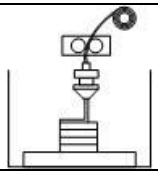
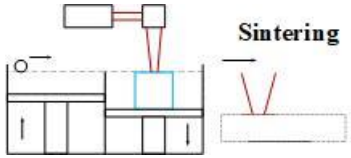
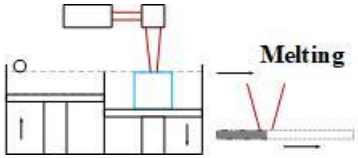
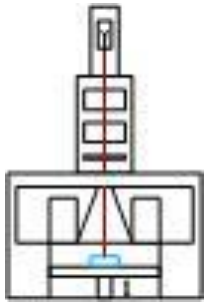
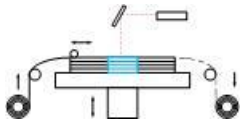
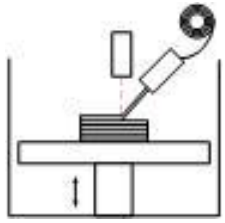


Figure 2.1 Mainstream AM technology categories.

Table 2.1 Overview of AM technologies.

Names of AM	Working Principle	Relevant Technologies	Materials	Advantages	Disadvantages	References
Vat Photopolymerisation (VPP) 	Photo-polymerisation and curing	**SLA**	Photopolymers	● High precision ● Good surface quality ● Fast prototyping	● Material limitations ● Complex post-processing ● Size constraint	(Pagac et al. 2021) (Davoudinejad 2021) (Al Rashid et al. 2021)
Material Jetting (MJT) 	A process associated with droplets of feedstock selectively deposited and successive layers cured.	Drop on-demand and nano-particle jetting.	Photopolymers, metals, wax	● High resolution ● Multi-material capability ● Superior surface finish	● High material costs ● Slow printing speed ● Support structure issue	(Sturm et al. 2019) (Yap et al. 2017) (Elkaseer et al. 2022)
Binder Jetting (BJT) 	A process selectively sprays liquid binder to bond powder materials	Inkjet 3D printing	Metal, ceramics	● Board material applicability ● High production efficiency ● High dimension accuracy	● Complex post-processing ● Material waste ● Lower strength	(Mostafaei et al. 2021) (Lores et al. 2019) (Lv et al. 2019)
Material Extrusion (MEX)	Heated polymer is extruded through a nozzle in filament form, which is then	**FDM**	Thermoplastics like PLA and ABS	● Diverse material options	● Lower precision ● Size limitations	(Huang et al. 2020)

Names of AM	Working Principle	Relevant Technologies	Materials	Advantages	Disadvantages	References
	deposited layer-by-layer onto a platform to create the 3D product.			• Low equipment cost • Easy operation	• Material Limitations	(Chaunier et al. 2018) (Goh et al. 2020)
Powder Bed Fusion (PBF)   	It is selectively fusing material by using laser beams or electron beams on the powder bed that can move upward and downward in the working area.	**SLS**	Thermoplastics	• Wide material applicability • No support structure needed • High production efficiency • Cost-effective	• Part shrinkage • Average surface quality • Material waste	(Dev Singh et al. 2021) (Singh et al. 2020)
		SLM	Titanium, stainless steel, aluminium	• High precision • High material utilisation • Can fabricate the complex structure • Superior part performance	• Low forming speed • High equipment cost • Needs post-processing • Support structure needs	(Vock et al. 2019) (Sun et al. 2021)

Names of AM	Working Principle	Relevant Technologies	Materials	Advantages	Disadvantages	References
		**EBM**	Most metal alloys (inc. titanium)	● High energy density ● Material strength ● Reduced residual stress ● High printing efficiency	● Not suitable for small holes and gaps ● High equipment costs and complex maintenance ● Limited printing size	
Sheet Lamination (SHL) 	The process joints sheets of material to form a part	Laminated object manufacturing	Paper, metal foils, polymer film	● Low cost ● Diverse material options ● Simple operation	● Lower strength ● Limited precision ● Material limitations	(Park et al. 2000) (Obikawa et al. 1999) (Bisht and Awasthi 2020)
Directed Energy Deposition (DED) 	A process exploits concentrated energy sources (laser, electron beam, or plasma arc) to melt the material being deposited.	**WAAM**	Aluminium alloys and steel	● Board material applicability ● High deposition efficiency ● Repair and modification capabilities	● High equipment cost ● Lower precision ● Large heat-affected zone	(Ahn 2021) (Gibson et al. 2021) (Liu et al. 2021b) (Tang et al. 2020)

2.2.2 SLS Technology

Selective Laser Sintering (SLS) is a process that utilises a laser to create layers of melted material. As shown in Figure 2.2, the powder is spread over the upper surface of the parts and heated to a temperature just below the sintering point by a powder roller. Once the powder reaches its melting point, a laser scanning system scans its cross-sectional contours (Sing et al. 2017). Subsequently, the powder is sintered and bonded to the lower layers. Afterwards, the print plate is lowered by a layer thickness and a layer of uniform and dense powder is applied to it by a roller (Frazier 2014). This sintering and powder distribution cycle is repeated until the entire build is completed (Ma et al. 2021). Nevertheless, when metal materials are combined with low melting point metals or polymers, different melting points can lead to porosity and poor mechanical properties, as the low melting point material melts during processing while the high melting point powder remains unaffected (Zhang et al. 2018a).

Despite these challenges, SLS has its merits in several aspects. Firstly, the wide range of materials can be utilised as the feedstock based on the product requirements. These materials include polymer, metals, ceramics, and sand. Moreover, high material utilisation is considered, and unused powder can be recycled for use in subsequent printing cycles (Paul and Anand 2012). In the SLS processes, the un-sintered powder serves as the support structure, thus eliminating the need for additional materials (Han et al. 2022). The mechanical properties of finished metal products are similar to those of conventionally manufactured metal products, which can be used for the fabrication of metal moulds and small-batch prototypes (Sing et al. 2017). However, the surface of SLS parts may be rough and require extra post-processing, which presents a challenge (Kumar 2003). Additionally, deformation can be observed when forming large-sized and high-performance metal and ceramic parts (Ma et al. 2018).

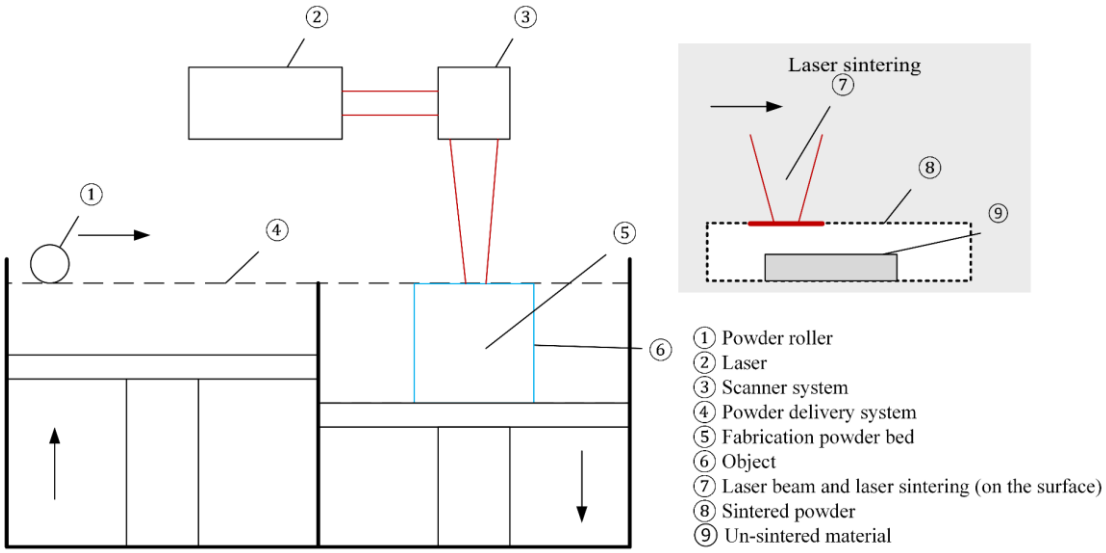


Figure 2.2 SLS system working principle.

2.2.3 Data Generation and Acquisition in AM Systems

AM has been recognised as a highly complex, integrated, and flexible system, including six critical stages during manufacturing, which are 1) **conversion**, 2) **positioning and orientation**, 3) **adding support structure**, 4) **slicing**, 5) **building**, and 6) **post-processing** (Ahuett-Garza and Kurfess 2018).

The conversion stage often involves transforming a Computer-Aided Design (CAD) model into a compatible file format to the 3D printers, which preserves the geometric information. This conversion is essential for the subsequent stages of the process. Due to its inherent simplicity, the STL format reduces the complex geometric information into a series of triangular facets, facilitating comprehension by design software and simplifying the data for 3D printing (Qin et al. 2019).

Positioning and orienting the prototypes is essential to placing and adjusting the converted models in the appropriate positions in the building envelope of an AM machine. This procedure contributes to the enhanced building efficiency and ensures

the mechanical integrity and surface finish of the prototypes (Wang and Alexander 2016).

Suspended components or features often need additional structures to support during the manufacturing process. On the other hand, the usage of support structures should be removed in the post-processing stage. This creation of support structures requires extended time and additional costs (Vaneker et al. 2020). Therefore, this step needs to be balanced when considering structural necessity and post-processing efficiency.

The slicing process converts a 3D model into a set of 2D contours. The process can print the products with these contours by following the generation of G-codes. G-codes can accurately guide and execute the manufacturing process of each layer for the 3D printers by determining process parameters such as layer thickness, print speed, and thermal settings (Zhang et al. 2023a).

Fabrication follows the G-codes and leverages different techniques such as deposition, solidification or fusion of materials, according to the specific machines and tasks (Dunaway et al. 2017). After prototypes are fabricated, manual intervention of post-processing is conducted to refine the prototypes. This is associated with the removal of support structures, finishing and curing (Bahnini et al. 2018).

In those procedures of producing builds, a variety of data is generated and utilised from the conceptual design to realisation (Chinchanikar and Shaikh 2022). Data are important to each stage, such as design-relevant data derived from CAD file models, STL files, geometry specifications, material attributes, process parameters and support structure (Mies et al. 2016). For instance, machine parameters such as power rate and thermal settings are critical for process control during the printing process (Wang et al. 2022b). These data can significantly influence the printing performance. On the other

hand, monitoring data from camera surveillance and inspection, provide critical information and insight for detecting distortion. This is helpful for quality control and identifying problems before printing (Zhang et al. 2023a). AM processes can benefit from continuous improvement in terms of AM processes by acquiring and analysing those data.

Overall, data from different sources such as working environment, machine settings, process operations, design specifications and material attributes play a vital role in analysing and optimising AM systems. The following section reviews the limitations and quality considerations of current AM processes to understand the challenges and opportunities in AM.

2.2.4 Challenges and Quality Considerations in AM

Growing demand for customers contributes to the recent advance in AM processes. These are driven by functional integration, design freedom and customisation (Fulga et al. 2017). By overviewing the development of AM systems, technologies, from resin-based processes such as SLA to more integrating techniques such as SLS and SLM, have broadened. Some critical technologies in energy sources such as lasers and electron beams are continuously optimised. The current AM machines have excelled at fabricating prototypes with more complex structures with enhanced precision (Thompson et al. 2016). AM refers to a more efficient production process in industries, but it cannot be employed in large-scale production. The reasons are:

- ***The Efficiency of the Production Mode***

The fixed workflow of AM consists of digital and physical processes, which convert conceptual designs to end products. A digital workflow incorporates operations such as data conversion, error checking, and slicing, providing instructions from a digital data stream. A physical workflow converts raw materials to final parts (Thompson et

al. 2016). Current 3D printers are not designed to endure long-term, high-intensity production loads (Pereira et al. 2019). The maintenance costs and complexity for a single 3D printer are significantly higher compared to traditional manufacturing processes (Kaikai et al. 2023).

- ***Material***

In the AM process, powder materials should be prepared in advance. In some instances, the mechanical properties of AM-produced parts are comparatively lower than those achieved through machining (Fulga et al. 2017). While titanium alloy components used in the aviation industry can meet mechanical property requirements, the overall performance of AM-produced parts remains questionable (Akilan and Velmurugan 2022). AM methods for metallic components, such as SLS and SLM, often result in poor surface quality and necessitate post-processing steps such as grinding and polishing (Srivastava et al. 2024). For 3D-printed parts with complex curved surfaces, removing support materials can be challenging and may risk damaging the final product.

- ***Costs***

Maintaining a single-machine operation under high workloads over the long term significantly increases the cost of large-scale production. The cost is also tied to the feedstocks used, which may not be cost-effective for highly customised production (Kanishka and Acherjee 2023).

Other constraints include 1) CAD and digitalisation (Hague et al. 2003), 2) support structures and build orientation (Leary et al. 2014), 3) process characteristics and machine capabilities, 4) metrology and quality control (Albakri et al. 2017), 5) through-life maintenance, repair and recycling (Campbell et al. 2013), and 6) external and regulatory constraints (Thompson et al. 2016).

Design for Additive Manufacturing (DfAM) can mitigate the negative impact of additively manufactured products, regulating and standardising the industry (Vaneker et al. 2020). It leverages various design approaches and tools to optimise functional performance and critical factors in the product lifecycle (Chinchanikar and Shaikh 2022). The term DfAM now refers to three levels: 1) tools, techniques, and guidelines, 2) understanding and assessing the impact of the design process on manufacturing performance, and 3) the relevance of design and manufacturing and its impact on designers, the design process and practice (Tang and Zhao 2016). In addition to the considerations mentioned in this section, sustainability raises another concern that is gaining the attention of academia and manufacturers. The next section illustrates various studies examining energy consumption and analytical methodologies in terms of impact factors of energy consumption and advanced data-driven modelling on energy. A multitude of optimisation and predictive modelling techniques have been reviewed.

2.3 Energy Consumption Analysis and Approaches in AM

The research focus on energy consumption prediction in AM has changed over time. It presents the development of technologies, environmental awareness, and the demand for sustainability practices. In the following section, a review of energy consumption analysis and approaches is demonstrated.

2.3.1 Early Recognition of Sustainability in AM

Sustainability in AM was recognised relatively later than other conventional manufacturing approaches. According to the literature, early studies have concentrated on understanding of environmental impact of AM systems. The analysis of energy consumption has been a highly active area of research in AM. Predictive modelling in AM is essential for understanding and improving energy consumption efficiency. These methodologies have utilised statistical analyses to examine the relationship

between various parameters and energy usage. For instance, Paul and Anand discovered that the correlation between the energy used in SLS and the total area sintered was determined by the thickness of the layer and the part orientation. According to their findings, the geometry of the part orientation affected the input energy, whereas the layer thickness was inversely proportional to the required energy (Paul and Anand 2012). A comparative analysis was carried out by Baumers et al. (2011) in terms of two laser-based systems by monitoring energy consumption to provide reliable data categorisation. Their research showed that material and process parameters, as well as geometry-related properties, influenced energy consumption. A further investigation on the influence of various factors on post-processing, such as the working environment, control parameters, part geometry, and machine settings, was conducted (Baumers et al. 2011).

Analytical research can be conducted early in the design and manufacturing process to facilitate environmentally friendly production (Niaki et al. 2019). For instance, Peng et al. (2018) discussed the sustainability of AM from a life cycle perspective and suggested focusing on the AM process and system (Peng et al. 2018). A study on different impact factors corresponding to energy consumption was identified in terms of four main data sources including working environment, design parameters, operations, and material attributes (Qin et al. 2018). In addition, Yang et al. (2017) extended the study to other different AM systems, which determined the impact of different manufacturing processes and energy consumption. On the other hand, the changing control parameters lead to a challenge due to different AM subprocesses (Yang et al. 2017). With ML-based approaches, energy prediction could be conducted to model the overall energy trend by identifying patterns in data, which is essential to optimise the process. Some authors have considered the data-driven approaches related to utilising ML.

2.3.2 Initial Energy Consumption Studies

Relevant research has switched to address AM energy consumption and its environmental influence more directly, which is associated with developing basic modelling and focusing on specific aspects of the manufacturing process. Some predictive models were employed to quantify the energy consumption usage according to the process parameters and stages of the printing processes. Yi et al. (2019) presented a simulation-based approach to model energy consumption in an SLM system. They also developed a five-phase analytical approach to examine energy consumption in the design phase, using a lattice ring fabrication as the case study (Yi et al. 2019). The exploration by Liu et al. (2018) broadened the research area in different systems in EBM and SLM, by considering the energy use of AM machine tools at both machine and process levels. They stressed high-energy beam generators, control systems, and cooling systems for AM machine tools, leading to a high influence on energy consumption (Liu et al. 2018b). In another PBF-based system, Ma et al. (2018) bridged the correlation between sintering parameters, energy consumption and material costs. According to their analysis, a non-dominated sorting genetic algorithm was employed to conduct a multi-objective optimisation on three variables including scanning speed, layer thickness and gap distance (Ma et al. 2018). In addition, Yan et al. (2022) leveraged a mathematical model to predict energy consumption. Some impact factors such as power, time and operating status are considered. Based on their finding, it demonstrated higher precision in terms of prediction, compared to energy-specific and process-based energy consumption models (Yan et al. 2022). According to a study on energy consumption modelling across FDM processes, Ma et al. (2021) utilised a mathematical model and analysed energy distribution profiles to construct the correlation between energy efficiency and performance enhancement (Ma et al. 2021). Dunaway et al. (2017) indicated that the surface area can affect the energy consumption in the FDM process (Dunaway et al. 2017). Tian et al. (2019) further revealed the quantitative relationship between energy consumption, part geometry and design parameters, by demonstrating a comprehensive framework within the FDM process (Tian et al. 2019a).

2.3.3 Energy Consumption Analysis and ML Integration in AM

Advanced techniques such as ML have started to be applied to predict energy consumption more precisely, aiming to capture and bridge complex relationships between process parameters and energy usage. Gutierrez-Osorio et al. (2019) conducted a comparative analysis of in-process predicted energy consumption based on part geometry (Gutierrez-Osorio et al. 2019). Yang et al. (2020) demonstrated energy consumption in an SLA process, using mathematical modelling and DL techniques (Yang et al. 2020a). Researchers such as Li et al. (2021) employed ML techniques to enhance the colour and quality of the deposited surfaces in a single-track titanium DED process. The researchers examined several process parameters, including laser power and scanning speed. In addition to these studies, there is a demand for more generic AM models that could be applied to a range of AM technologies. However, the trade-off between model complexity and practical usability remains to be explored. Model complexity might pose a challenge when computational resources are limited. It means that simpler models would be more appropriate for industrial environments but not capture the full range of variables that affect energy consumption.

Owing to advanced DL in AM, energy efficiency can be enhanced through dynamic adjustment of process parameters swiftly and accurately. The non-linear relationship between various parameters and energy consumption increases the complexity of data analytics and modelling (Fu et al. 2022). DL models are suitable for capturing and revealing non-linear relationships more effectively. Additionally, it is worth noting that the dynamic working environment of AM systems is driven by operational and process parameters, significantly influencing energy consumption (Thompson et al. 2016). The next section will focus on DL-based approaches in energy consumption predictive modelling.

2.3.4 Advanced Prediction Models and Holistic Energy Optimisation in AM

- ***DL Applications in AM Energy Consumption Predictive Modelling***

More recent studies have focused on more complex and robust DL models to predict energy consumption, where DL can uncover hidden patterns and insights from data and energy usage. It allows the AM to adjust and correct processes in real-time scenarios, which could optimise designs and overall processes (Rai et al. 2021). In the field of robotic AM, Ghungrad and Haghighi (2024) introduced a multi-point trajectory-based energy consumption model. They developed two alternative models for real-time energy consumption prediction: a purely data-driven model and a kinematic-based data-driven model. These models improved prediction accuracy by learning inverse kinematic solutions on trajectories and showcased their potential in real-time applications in practical case studies (Ghungrad and Haghighi 2024). Wang et al. (2024) demonstrated that advanced pattern matching can significantly reduce the cost of training data. The study was based on recognising filling patterns of simple structures and arbitrary shapes to approximate energy consumption by employing dynamic algorithms (Wang et al. 2024c). Lim et al. (2021) employed DL to optimise the deposition surface colour and quality in a DED process. The parameters included laser power and scanning speed (Lim et al. 2021). Another hybrid DL approach was carried out by Hu et al. (2021). In their study, a CNN-LSTM model fused multi-source data to predict energy consumption, revealing the correlation between operational environment parameters and energy consumption in an SLS system (Hu et al. 2021a). In another laser-based PBF system, Ghansiyal et al. (2023) employed a conceptual framework to forecast the layer quality and the energy required to build the layers by using the proposed multimodal regression method by integrating 2D images and process parameters (Ghansiyal et al. 2023).

DL has the ability to solve the challenges posed by increased data complexity (Tian et al. 2019a). Wang et al. (2022) developed a new continual attention memory network to predict energy consumption. In detail, their work was based on extracting and storing complementary information in the FDM system, in which the inherent consistency between layers contained more diverse and valuable insights (Wang et al. 2022a). In another study, El youbi El idrissi et al. (2023) introduced a DL-based method by using a Multilayer Perceptron (MLP) network, and they identified the

relationships between energy consumption and impact factors, including part orientation in the FDM system (El youbi El idrissi et al. 2023). By generating filling patterns and trajectory planning with generative adversarial networks, Xu et al. (2020) converted 3D objects into different layers in an FDM process. This adaptive multi-layer customisation leads to improved energy efficiency, linking it to customisation parameters, according to their findings (Xu et al. 2020).

- ***Predictive Modelling for Energy Optimisation Support in AM***

Beyond the scope of energy prediction, current research has focused on holistic optimisation support to consider both energy consumption level and in-process parameters, preserving energy efficiency. The predictive modelling in AM has shown the potential to improve energy management and optimisation (Niaki et al. 2019). Ulkir (2023) established a Life Cycle Assessment (LCA) based on a mathematical model to assess overall environmental impacts over the life cycle of AM products, which included resource efficiency and waste generation (Ulkir 2023). By quantifying energy consumption, Gao et al. (2024) proposed a comprehensive analytical approach for complex AM solutions. They divided parts into typical features, creating energy models for each feature (Gao et al. 2024). In addition, a new mathematic model was proposed by Yan et al. (2022) to predict energy consumption by considering the power of each component, the time of each process and the operating state of each component, demonstrating that it had higher prediction accuracy compared to energy-specific models and process-based energy consumption model (Yan et al. 2022). Consequently, a DL-based study has been conducted to optimise the deposition surface colour and quality in the single-track titanium alloy DED process.

Hasan et al. (2023) studied the effect of process parameters such as scanning speed, laser power and feed rate on energy consumption. The findings indicated that laser power has a significant influence on energy consumption and therefore higher scanning speeds, lower laser power and feed rates are recommended to improve energy efficiency. This research is critical to understanding and optimising energy usage in

advanced manufacturing processes (Hasan et al. 2023). Another study carried out by Tiwari and Yang (2023) was based on the energy consumption behaviour of 3D printed Carbon Fibre Reinforced Polymer (CFRP) parts. They developed an energy model during the melt and deposition phases and identified that layer height, filler density and extruder speed were the most important factors impacting the energy consumption behaviour. The insight offered a sustainable decision support tool for CFRP design and process planning (Tiwari and Yang 2023). In addition, Liu et al. (2021) proposed a decision model that compared the energy consumption in additive subtractive hybrid manufacturing with conventional manufacturing. They focused on the entire lifecycle of a printed prototype before and during the manufacturing process. This study determined that components utilised in the aerospace industry were more compatible with the ASHM process, due to the lightweight feature and high Energy consumption Reduction Coefficient (ERC), leading to reduced energy consumption in the operational phase compared to conventional processes (Liu et al. 2021a).

This section reviewed the development of establishing predictive modelling in different AM systems to understand and optimise energy efficiency, using DL approaches. The subsequent section will explore advanced DL techniques for data analytics in AM systems.

2.4 Advanced Data Analytics in AM

In the previous section, the DL techniques for energy consumption modelling in AM were reviewed. This section will review the advanced data analytics in AM systems in terms of other areas. This section is organised as follows:

In Section 2.4.1, the background of DL will be described. Classical architectures such as Fully Connected Neural Networks (FCNNs), Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs) are the most fundamental DL architectures, which will be reviewed as well. Section 2.4.2 will review the recent

advances in DL for AM systems. At last, the challenges and opportunities associated with current DL-based approaches will be discussed (Section 2.4.3).

2.4.1 Background of DL

In 2006, Deep Learning (DL) was first introduced based on the concept of Artificial Neural Networks (ANNs) (Hinton et al. 2006). Compared to traditional approaches to AI and ML, DL concentrates on model efficiency and compactness. With a computational model consisting of many hidden layers, DL can learn data representations with multiple abstraction levels (LeCun et al. 2015). DL is inspired by the interest in creating and mimicking neural nets in the human brain for analysis and understanding. DL has been identified to extract features without supervision and aims to model high-level data abstractions (Gheisari et al. 2017). ML is limited when solving problems involving signals such as human speech and raw images, as well as problems involving complex classification because of generalisation. On the other hand, when it comes to simulating more complicated functions that require larger training datasets, DL is superior due to its non-linear and deep architecture (Goodfellow et al. 2016). The relationship between AI, ML and DL is demonstrated in Figure 2.3.

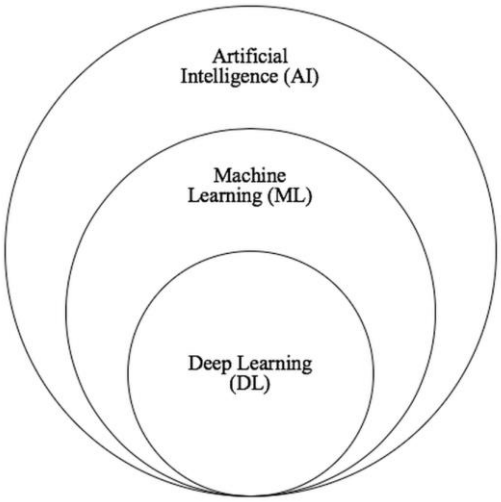


Figure 2.3 The relationship between AI, ML and DL.

DL technologies have become a popular topic in ML, AI, data science and analytics due to their ability to learn from and generalise unseen data. Depending on the different datasets, DL can address regression and classification problems, yield significant results and uncover hidden knowledge and inherent correlations between data (Sarker 2021). DL can perform a variety of tasks such as image recognition, video classification, text generation and speech processing (Minar and Naher 2018). There are three most basic categories in DL techniques based on their architectural designs, including 1) deep architecture such as FCNNs or DNNs, 2) CNNs, and 3) RNNs. In the following sections, these different DL architectures will be reviewed and discussed.

- ***Fully Connected Neural Network***

Fully Connected Neural Networks (FCNNs), or known as **Dense Neural Networks (DNNs)**, are characterised by having multiple layers of linear and non-linear operations. These networks are capable of approximating complex functions that map input data to outputs (Wang et al. 2020b). Different from standard NNs, DNNs extend the depth by adding hidden layers, which allows the model to learn more complex and abstract representations of the raw input data. This depth is critical for managing the complexity of the related learning tasks (Janiesch et al. 2021). Figure 2.4 illustrates a typical FCNN architecture, demonstrating the data flow through layers and the transformation from the input to output via a series of interconnected neurons.

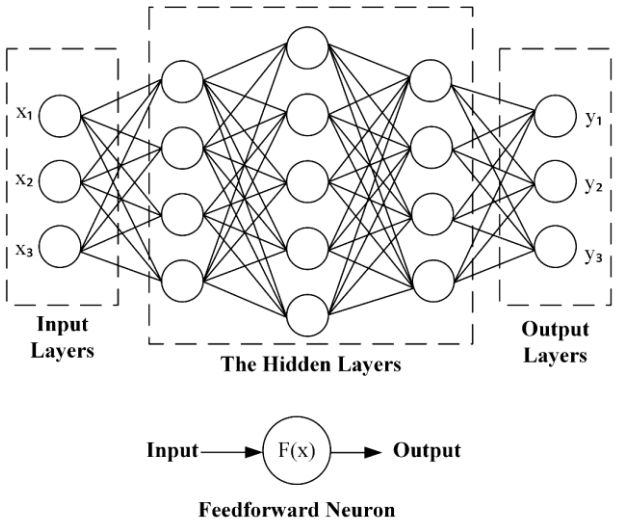


Figure 2.4 FCNN architecture overview.

- ***Convolutional Neural Network***

Convolutional Neural Networks (CNNs) are a variant of DNNs, which learn directly from the structural data without human intervention in feature extractions (Minar and Naher 2018). In addition, CNNs have been tailored to process image data, typically accepting input with three dimensions: height and width of images and colour channels in RGB (Liu et al. 2018a).

In Figure 2.5, a typical CNN architecture consists of convolutional layers, down-sampling (pooling) layers and fully connected layers. In addition, CNNs can employ a dropout technique to mitigate overfitting, different from traditional network architectures (Sarker 2021). Convolution plays an important role in extracting features from the input image and generating a feature map as output. The convolutional layer converts the image into a series of values that allow network nodes to interpret and extract hidden patterns from edges and textures (Albawi et al. 2017). The convolutional layer includes multiple convolution kernels that compute various feature maps, where each neuron in a feature map is connected to a group of neighbouring neurons from the preceding layer. It allows the network to detect hidden patterns with spatial hierarchies (Gu et al. 2018). The pooling operation is subsequently employed

to reduce redundancy since the considerable image input leads to overfitting, thereby reducing the spatial size of the representation to decrease the number of parameters (i.e. weights and biases) and computations in the network (Li et al. 2022). The fully connected layer performs a similar role as ANN. It offers classification scores that can be used for classification by integrating learned features from previous layers to make a final prediction (Saxena 2022). CNNs have promising applications such as time series predictions and signal identification for one-dimensional CNNs. In addition, two-dimensional CNNs show the merits of image classification, object detection, image segmentation and face recognition (Li et al. 2022).

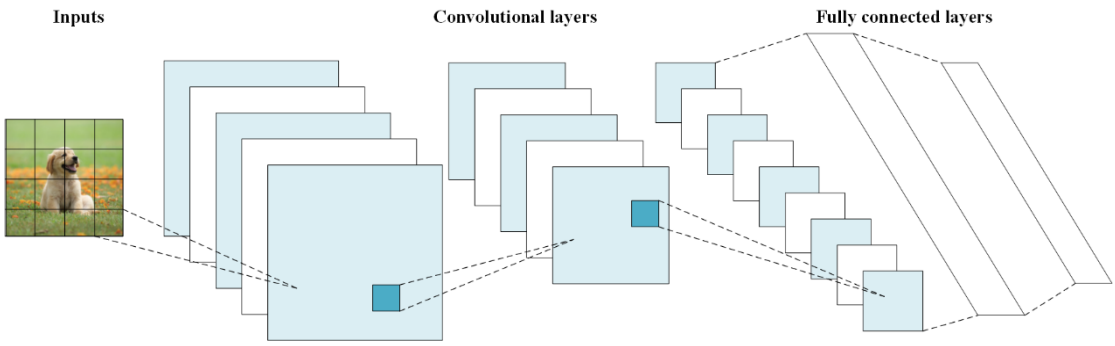


Figure 2.5 CNN architecture overview.

- ***Recurrent Neural Network***

Several real-world problems involve sequential data, such as text, speech, and video. However, the correlation between data has not been considered in the feedforward neural networks, and the network output is only relevant to the inputs at the current moment, which limits their effectiveness for sequential data analysis (Salehinejad et al. 2017). As shown in Figure 2.6, ANNs with multiple recurrent connections, **Recurrent Neural Networks (RNNs)** are utilised to process time series data or sequential data, allowing for the analysis and prediction of data where order and timing are crucial, which offer the solutions to ordinal or temporal problems, including language translation, natural language processing and text generation (Sarker 2021). A key distinguishing factor is their capacity to leverage past inputs to affect current inputs

and outputs, known as the “memory” (Sutskever et al. 2011). Instead of strictly memorising all fixed-length sequences, RNNs store previous time-step information by hidden states, enabling them to handle sequences of arbitrary lengths with a fixed number of parameters (Pascanu et al. 2014).

Unlike conventional DNNs, the nodes in each layer of an RNN are connected via a loop, enabling the network to propagate information from one step to the next. This self-connection enables RNNs to preserve information over time within a sequence of data (Wang et al. 2020b). This allows for more efficient use of the parameters. RNNs share parameters in each layer, which have different weights for each node. In addition, RNNs employ the same weight parameters within each layer (LeCun et al. 2015). Bidirectional Recurrent Neural Networks (BRNNs) to enhance contextual understanding, Long-Short Term Memory (LSTM) to address the vanishing gradient issue, and Gated Recurrent Units (GRUs) as a simplified alternative featuring forget and input gates, are the variants of RNN architecture (Salehinejad et al. 2017). These variants can model dynamic systems, providing a more effective way to manage sequential or time-series data.

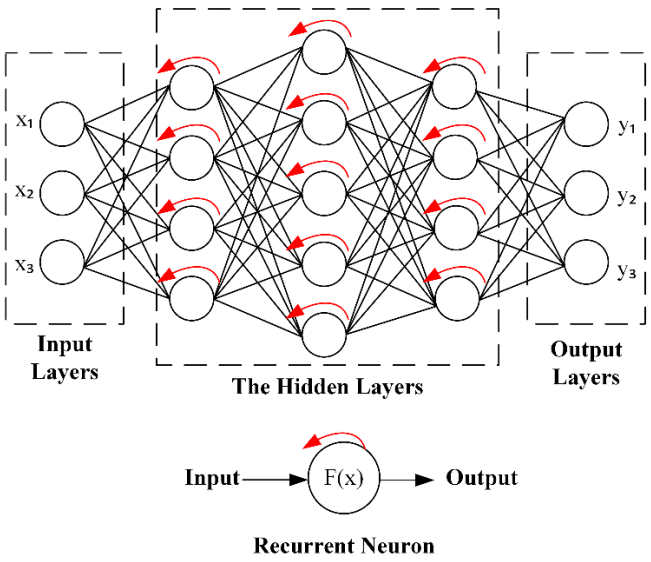


Figure 2.6 RNN architecture overview.

2.4.2 Recent Advances in DL in AM

The complexity of AM systems involves various technologies including computer science, materials science, mechanical engineering, and electronics engineering. Mapping the correlations between different variables at different stages of development using mathematical relationships is challenging. In addition to energy modelling mentioned in Section 2.3, advances in DL have profoundly influenced AM in other aspects such as material development, process optimisation, quality control, and design optimisation.

- ***Material Development***

DL technologies have a wide range of applications in material design discovery and manufacturing (Papadimitriou et al. 2024). DL models can be utilised to predict the properties of new materials and materials combinations, thereby accelerating the discovery of materials optimised for 3D printing (Jin et al. 2020). Zuccarini et al. combined their experiment and DL technology to predict and synthesise new materials with potential application value, holding a better prediction accuracy and efficiency than those of conventional approaches, which provides a new pathway to explore new materials (Zuccarini et al. 2024). Erps et al. expanded this application by developing multi-objective optimisation algorithms based on ML. This technique can identify the optimal solution in a quick manner in terms of the complicated space of 3D printed material, thereby shortening the material development cycle (Erps et al. 2021). Different from traditional and physical-based approaches, the CNN provides superior prediction performance by learning complex patterns. Another study stressed the effect of re-entrant honeycomb structure design, using a DL-based approach. Instead of leveraging conventional complex analytical equations and control algorithms, the proposed approach leveraged pseudo-randomised images of geometric modifications to develop the prediction model for deviations from multi-material configurations (Wilt et al. 2020).

- ***Process Optimisation***

Ng et al. (2024) believed that the DL algorithm played an essential role in optimising print parameters by analysing data in the process. According to the different materials used, adjustment of laser power, print speed and layer thickness were considered when considering a metal AM process. Furthermore, extrusion speed and temperature were involved in the case of filament-based printing. These optimisations on printing parameters can improve the strength, precision and surface finish of the prototypes (Ng et al. 2024).

Many researchers have investigated methods to enhance product quality. Nohut and Schwentenwein (2024) leveraged ML technologies to optimise the manufacturing process of multi-material ceramics by predicting shrinkage and porosity (Nohut and Schwentenwein 2024). A surrogate model based on a feedforward neural network was developed by Pham et al. (2023) to predict temperature changes and melt poo size in AM process, which reduced the dependence on costly computational simulations (Pham et al. 2023). Tamir et al. (2023) proposed an open-loop and closed-loop control to monitor the impact of processing parameters on the quality of printed objects. By integrating an open-loop control with a fuzzy algorithm, a closed-loop control algorithm was developed. According to their results, this method determined the correlation between 3D printing specifications and the in-process parameters (Tamir et al. 2023). A more recent study explored the integration of point clouds and DL to enhance in-situ quality through data analysis and intelligent decision-making. As a result of utilising a deep convolution autoencoder model to process a top-surface point cloud, a statistical analysis of the part quality was conducted, which was necessary to generate commands for the optimisation of the manufacturing process. Since the G-code has been modified, the auto-encoder results can be employed to adjust layer deposition in real-time, as controllers leveraged layer deposition grid-by-grid by adjusting the feed rate and print speed (Akhavan et al. 2024).

- ***Design Optimisation***

Utilising various AM techniques, engineering research aims to develop multifunctional intelligent composites. There are limitations to large-scale industrial applications of AM due to the lack of reliable methods to predict and model material properties, design barriers, limited material libraries, processing defects, and inconsistent product quality (Babu et al. 2023). Topology optimisation includes a broad concept of determining optimal layouts for structural materials based on computational analysis. Discrete optimisation techniques have consistently been used in civil and structural engineering. In contrast, optimisation of continuity has recently emerged as a powerful tool to promote the adoption of AM, as observed in several other fields of industry (Ribeiro et al. 2021). In their study, a comprehensive analysis of structural steel design and AM-specific design was carried out during the discussion of recent DL-based approaches and fields of application of topology optimisation.

An in-depth analysis of the heterogeneous graph structures of spider webs allowed the researchers to model and synthesise artificial, bioinspired 3D web structures using DL. Generative models consider critical geometric parameters such as edge length, node number, and averaging node degree. As a result of inductive representation sampling of large experimentally determined spider web graphs, this study uncovered graph construction principles, which could be used to inform the training of conditional generative models (Lu et al. 2023b). The textured surface on tribological performance was considered by Zhu et al. (2023). Their focus is on the development of a DL-based generative design framework integrated with CNNs and an enhanced Monte Carlo search. Based on their findings, they indicate that machine-generated wavy and chevron-like textures can enhance tribological performances in terms of sliding surfaces with infinite design domains (Zhu et al. 2023). Wu et al. (2023) developed an innovative multiscale topology optimisation technique leveraging a derivative-aware DL algorithm to achieve uniform strain patterns on additively manufactured lattices (Wu et al. 2023a). Another study focused on layered surface morphology information, according to Liu et al. (2023), who presented a CNN-based approach to solving high-dimensional and nonlinear problems in 2D images and 3D point clouds. The trained

model could directly predict 3D surface data, thus reducing time-consuming triangulation calculations (Liu et al. 2023).

- ***Quality Control***

While handcrafted-based approaches can extract representative features of images, they can be challenging when dealing with complex design and structural features produced by different AM processes. DL can be utilised in AM systems to provide predictive insights for identifying complex manufacturing patterns, and it would allow the system to make intelligent decisions in manufacturing (Jiang et al. 2022). DL in metal AM hold the potential to optimise manufacturing processes and improve part quality (Johnson et al. 2020).

The research focuses on the application of DL in AM in terms of quality control and monitoring. Fischer et al proposed a DL-based approach for monitoring and classifying powder bed defects in metal AM. Their study employed the Xception model to classify powder bed images, identifying defects with high accuracy (Fischer et al. 2022). Manivannan exploited semi-supervised DL to achieve automatic quality inspection in AM, which was validated on multiple AM datasets demonstrating excellent performance with less annotation data (Manivannan 2023). Li et al. (2020) proposed a DL-based quality recognition method for metal AM, which mitigated the high demands of high-quality annotation data by exploring semi-supervised training data (Li et al. 2020). Lu et al. (2023) developed a real-time system to detect defective areas in printed objects, by integrating DL models with geometric analysis. In addition, they quantified the severity of each unique defect based on the misalignment level (Lu et al. 2023a). Another DL-based defect detection approach employed three YOLO attention mechanisms for further improvement in the performance of the model (Li et al. 2023). In addition, Kumar et al. (2024) presented a comprehensive study on the DL approach in defect detection using zero-bias DNN with less manual image processing. They focused on the feasibility of detecting multiple types of defects, including cracks, stringers and warpages. These data did not require a priori knowledge from trained

datasets (Kumar et al. 2024). A study conducted by Fang et al. (2024) highlighted the importance of deep transfer learning in detecting outliers in a WAAM process. When a new domain adaptation strategy was designed using the Particle Swarm Optimisation (PSO) technique, the study minimised the cross-domain discrepancies between marginal and conditional distributions to mitigate data imbalances (Fang et al. 2024).

2.4.3 Challenges and Opportunities in DL for AM

Traditional statistical approaches have become inadequate when managing numerous datasets to extract valuable information from heterogeneous data produced by AM systems (LeCun 2019).To address this problem, complex architecture with more hidden layers in DL architecture plays an important role, which leads to intensive parameters. These analytical models can often extract and learn valuable insights from data (Yang et al. 2020b). For instance, an ensemble model can achieve the tasks of discovering the hidden knowledge, but it fails to deploy on edge devices due to slow inference speed and high resource requirements (Lin et al. 2020). This step requires a lot of latency and computational resources (Gou et al. 2021). As a result, model compression approximates the performance of a slower, more complex, but more accurate model with one faster and more compact (Schohl 2003), which involves reducing the number of parameters while retaining performance.

As one of the prevailing architectures applied in various scenarios, CNNs take image data directly as input without requiring complex operations such as additional manual image preprocessing and feature engineering. This type of architecture uncovers hidden patterns in the image data by extracting representative features for sophisticated shapes (Saxena 2022). According to the research, CNN is the primary tool for automatically learning and extracting representative features from highly complex geometries. In order to achieve higher performance on feature extraction, it is necessary to significantly increase the number of layers and their parameters (Hu et al. 2021b). Despite the advantages, convolution operations are always complex and time-consuming due to their high computation requirements, which convolution accounts

for approximately 90% of computation time (Cong and Xiao 2014). Therefore, the complexity and memory requirements of the model lead to the limited usage of deep models on resource-constrained devices in the manufacturing process. Despite the high level of performance achieved by current deep models, their implementation on the edge platform is constrained by high latency and memory requirements (Wang et al. 2020a). These models with a deep architecture are computationally intensive and consume a substantial number of resources, making them challenging to deploy, particularly in applications deployed on devices with limited resources (Wang and Yoon 2020). To overcome these limitations, model compression techniques are employed, including network pruning, quantisation, low-rank approximation, and Knowledge Distillation (KD) (Cheng et al. 2018).

2.5 Model Compression Techniques

Numerous real-world applications require devices with real-time processing capabilities. Since the current DL-based models are computationally intensive, the main obstacle to using them is the limited resources of edge devices. Limited memory and computational power pose challenges in deploying DL models effectively on resource-constrained platforms. As the model complexity is directly proportional to its storage requirements, directly deploying it on devices with limited resources is challenging. Furthermore, larger models demand longer inference times and higher power consumption (Cheng et al. 2017). Consequently, model compression techniques are implemented. The following sections will review four model compression techniques: pruning, quantisation, low-rank approximation, and Knowledge Distillation (KD). The following sections are organised as follows: Section 2.5.1 involves the pruning technique for compressing model. Section 2.5.2 will review the quantisation technique, followed by low-rank approximation in Section 2.5.3. Section 2.5.4 will focus on KD techniques.

2.5.1 Pruning

The purpose of pruning techniques is to identify redundant connections and remove them so that they are not involved in the forward or backward operations of the network, which reduces the amount of network computation (Wang et al. 2024a). The neurons and connections that have been removed are no longer retained, resulting in a reduction of storage in the model. At the end of this process, a network, that is initially dense, becomes sparse due to the removal of specific connections (Liu et al. 2018a). Figure 2.7 depicts the process of network pruning.

One of the most challenging aspects when pruning is to identify the less critical parameters. Pruning networks leads to a reduction in network complexity and mitigates overfitting. The architecture of a larger neural network is pruned to yield a less complex neural network. Pruning employs a top-down approach, wherein a large network is first constructed, and subsequently, the network structure is trained by removing or merging specific neurons or weights as necessary (Reed 1993). By setting weights to zero, magnitude-based weight pruning results in the sparsity of the model during the training process. The pruning technique sets weights below a certain threshold to zero by comparing their absolute values against the threshold. As part of this approach, knowledge of connectivity must first be acquired through training (Han et al. 2015).

Unlike conventional training, the connections are learned instead of learning the final value of weights. Subsequently, low-magnitude neurons are pruned out, which transfers a dense neural network to a sparse one, removing all connections with weights below the threshold. Finally, once the network is retrained, a final weight is calculated for the remaining sparse connections to preserve the accuracy (Cheng et al. 2018). Some significant branches of pruning include weight pruning and filter pruning.

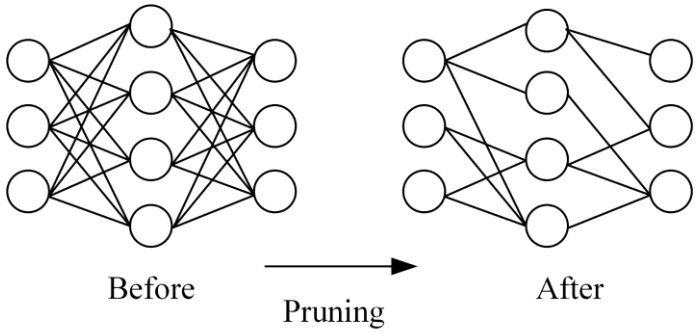


Figure 2.7 Pruning technique.

2.5.2 Quantisation

Quantisation in DL refers to representing the parameters of an NN using floating point numbers with low-bit width integers, as shown in Figure 2.8. Quantisation involves clustering the weights of neurons. It maps the range of these weights to the INT8 integer scale, ranging from -127 to 128, using the maximum and minimum values found in the data (Choudhary et al. 2020). Through this method, on the one hand, the storage of the model can be reduced. The weight parameters of the model are often stored in the form of 32-bit floating-point numbers with a considerable quantity, consuming a large storage space (Gou et al. 2021). When the number of parameters is reduced, the model will be reduced when the 32-bit floating-point number is quantised into an 8-bit fixed-point number (Li et al. 2022). Post-quantisation model reduction also leads to a significant decrease in computational resources needed during the network's forward computation phase (Polino et al. 2018). Additionally, employing fewer bits per weight decreases the data required for computations, resulting in energy savings and reduced access costs (Cheng et al. 2018).

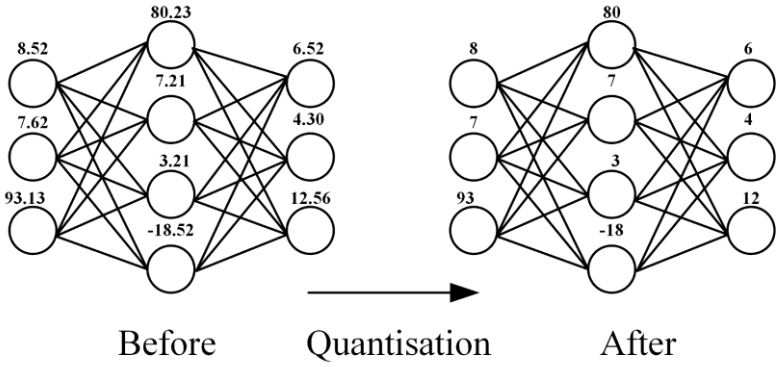


Figure 2.8 Quantisation technique.

As a result of quantisation, fewer bits are required to represent weights and biases in a CNN, reducing storage and memory requirements and computation complexity.

$$Q(v) = round\left(\frac{v}{S}\right) + Z$$
$$where\ S = \frac{2^b - 1}{\alpha - \beta}$$
$$and\ Z = -round(\beta \cdot S) - 2^{b-1} \quad (2.1)$$

A floating-point weight or activation can be converted into an integer using the equation, where $\boldsymbol{Q}$ denotes the quantised value,$\boldsymbol{v}$ is the original floating-point value, $\boldsymbol{S}$ is the scaling factor, and $\boldsymbol{Z}$ is zero point to ensure that the quantised integer value represents zero correctly. The actual values are located at the range of $\boldsymbol{[\beta, \alpha]}$, which is at the range of $\boldsymbol{[-2^{b-1}, 2^{b-1} - 1]}$ (Jacob et al. 2018; Wu et al. 2020). Equation (2.2) and Equation (2.3) show quantisation and de-quantisation, respectively, where $\hat{X} \approx X$.

$$X_q = clip\Big(round(X * 2^{b-1})\Big) \quad (2.2)$$

$$\hat{X} = \frac{1}{S}(X_q - Z) \tag{2.3}$$

2.5.3 Low-Rank Approximation

The low-rank approximation-based approach enhances the computational process of the model from the perspective of decomposing matrix operations, greatly reduces model redundancy, and significantly speeds up model computation when compressing and accelerating the fully connected layers (Cheng et al. 2017). Most of the computation occurs in the convolutional layer, whose parameters are usually stored as multidimensional matrices. Decomposing the matrix into a series of smaller matrices through linear algebra allows the combined smaller matrices to approximate the representation of the original convolutional layer (Choudhary et al. 2020). This model compression technique can preserve the model's accuracy while decreasing the demand for parameter storage requirements. The low-rank approximation is performed starting from the shallowest layer to the deepest layer, achieving the approximation at each convolutional layer. After the decomposition of a layer, its parameters are fixed and fine-tuned (Dziugaite and Roy 2015).

2.5.4 Knowledge Distillation

The concept of model compression via **Knowledge Distillation (KD)** was first proposed by Buciluǎ et al. in 2006 (Buciluǎ et al. 2006). After that, Hinton et al. (2015) systematically defined and introduced the training approach of the KD technique (Hinton et al. 2015). KD involves training a compact model to imitate a pre-trained large model (or ensemble of models) that has been previously trained (Ba and Caruana 2013). Many models have more recently achieved State-of-the-Art (SOTA) performance with current algorithms. As a result of excessive latency and memory use, the model structure makes it computationally expensive and inefficient.

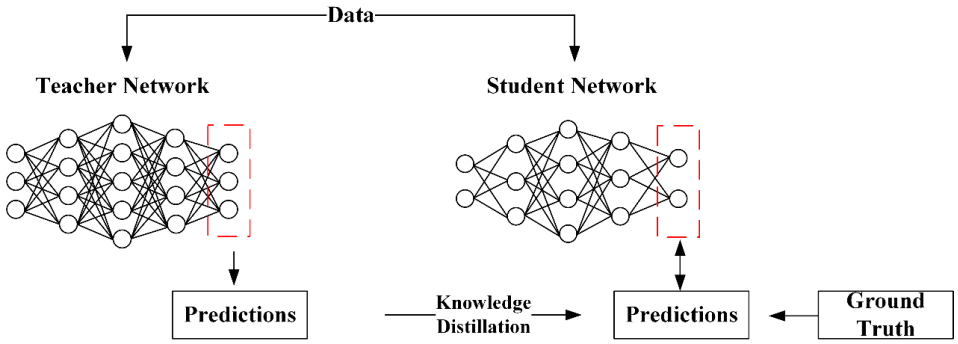


Figure 2.9 Knowledge distillation in teacher-student architecture.

Figure 2.9 illustrates a teacher-student architecture in the distillation process, where the cumbersome model obtained during the training phase represents a teacher and the distilled model represents a student. Students are trained by teacher networks with high learning capacity, transferring knowledge to student networks with lower learning capacity, thus enhancing the model's generalisation ability (Hinton et al. 2015). The term "knowledge" is a mapping from input to output vectors. The class possibility output from the teacher model serves as labels for the data, sent to the student model for training, with a soft target representing class probabilities. The distilled model is trained on a dataset, employing a soft target distribution for each instance in the dataset.

In contrast to the original softmax function, Equation (2.4) introduces a temperature hyperparameter, ***T*** to smooth the probability distribution between teacher and student models.

$$q_i = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_i \exp\left(\frac{z_j}{T}\right)} \tag{2.4}$$

Equation (2.5) identifies the distillation loss, including soft label ($\boldsymbol{q_i}$) of the teacher model, and the class possibility ($\boldsymbol{p_i}$), and temperature parameter ($\boldsymbol{T}$).

$$L = T^2 \sum_i^N q_i^T \log(p_i^T) \tag{2.5}$$

Equation (2.6) represents the aggregation of soft labels, where $\boldsymbol{w_{ij}}$ are the weights assigned to the soft labels $\boldsymbol{q_{ij}}$ from the teacher model.

$$q_i = \sum_j w_{ij}\, q_{ij} \tag{2.6}$$

Hard label loss and soft label loss are added to compute students' loss functions (Equation 2.7), which are derived from cross-entropy, derived from the probability distribution of output, and label for student networks, and Kullback-Leibler (KL) divergence, which is equivalent to the difference in output between student and trained teacher networks, respectively. $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are used to control the ratio between two losses.

$$L_{student} = \alpha L_{soft} + \beta L_{hard} \tag{2.7}$$

KD can be classified into three categories, logit-based, feature-based and relation-based, according to their knowledge types. A special focus will be placed on the development of teacher-student architectures, which are crucial for KD in DL. Learning from large networks is challenging due to the significant model capacity gap between the teacher and student models. A variety of methods has been developed to ensure knowledge is effectively transferred to student networks.

- ***Logit-based Knowledge Distillation***

In Figure 2.10, **logit-based** or **response-based** knowledge refers to the neural response to the final output of the teacher model (Gou et al. 2021). Owing to its simplicity and effectiveness, this KD technique is used in a wide variety of tasks and applications. A typical KD process consists of three components: a teacher network, a student network and knowledge transfer between them. The first detailed studies of KD have focused on learning category distributions, i.e., using soft labels from large pre-trained teachers to train small student models. The student network took the output of the teacher network as input and aimed to approximate the output of the teacher network (Hinton et al. 2015). In the traditional KD process, knowledge is transferred from a complex model to a simplified model. Zhang et al. (2018) introduced a mutual learning strategy, where a group of student models learned and guided each other, replacing the classic teacher-student architecture. The results of the study showed that model performance tended to improve as the number of student models increased (Zhang et al. 2018b).

Soft labels can enhance classification tasks by providing more information and revealing the teacher model's generalisation ability. However, Cho and Hariharan (2019) found that a better teacher model did not necessarily lead to improved student performance when using soft labels with regularisation. Student models did not match the teacher's performance due to the potential for mismatched samples and a bias towards primary losses (Cho and Hariharan 2019). Xie et al. (2020) presented a logit-based KD, using more noisy datasets to enhance the student model performance by focusing on the data issue (Xie et al. 2020). Yang et al. (2019) utilised soft labels when training the student model, optimising it by applying constraints. It has been shown that combining ground truth with the secondary class leads to more effective learning, which prevents overfitting (Yang et al. 2019). However, this teacher-student architecture was limited by structural differences between teacher and student models. Aiming on this issue, Phuong and Lampert (2019) proposed a new loss function and a multi-exit architecture for the KD technique, which utilised an early exit to mimic the

more accurate later exit by matching their output probabilities (Phuong and Lampert 2019).

The logit-based method is relatively simple and effective due to its straightforward approach and strong performance. Utilising a teacher model in student modelling provides probability distributions that serve as similarity information and additional supervision, facilitating student learning (Gou et al. 2021). The logit-based KD techniques face several challenges. For instance, logit-based KD is sensitive to the temperature hyperparameter, which can negatively affect the performance of distillation (Hinton et al. 2015). Moreover, logit-based distillation focuses on the final output, which neglects the intermediate representation, leading to the loss of key insights (Gou et al. 2021).

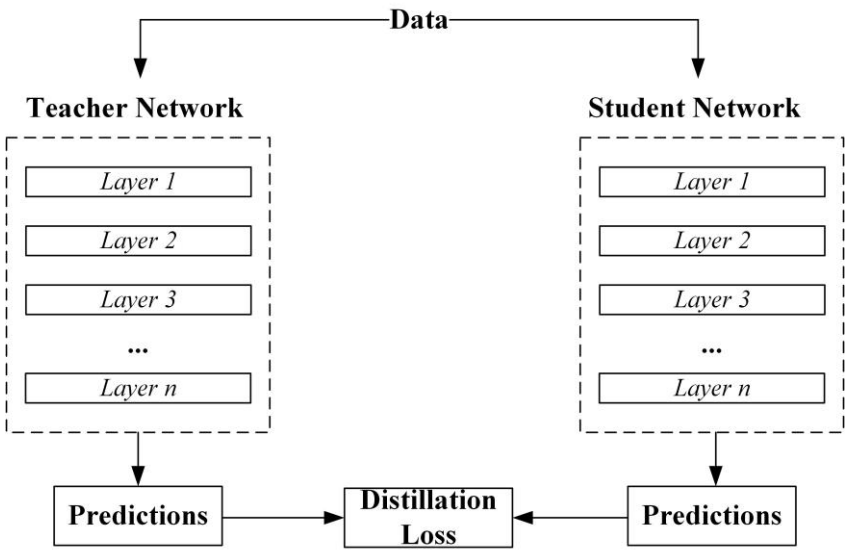


Figure 2.10 Logit (response)-based KD architecture.

- ***Feature-based Knowledge Distillation***

Figure 2.11 depicts the **feature-based** KD. This strategy extracts features from the intermediate hidden layers in the architecture of the teacher model (Gou et al. 2021). Romero et al. (2015) first introduced the idea of feature-based KD, which employed the teacher output in its hidden layer to supervise students. The technique directly matched the feature activations of the teacher and the student (Romero et al. 2015). Inspired by this, the method was proposed to match the features indirectly. According to Zagoruyko and Komodakis (2016), an attention map was derived from the original feature maps to express knowledge (Zagoruyko and Komodakis 2016). Chen et al. (2021) developed a new teacher-student architecture, by utilising a locality-preserved loss function. This loss function allowed the student network to generate low-dimensional features from high-dimensional features of the teacher network (Chen et al. 2021). Another Dual Masked Knowledge Distillation (DMKD) scheme was proposed by Yang et al. (2024), which employed a dual attention mechanism for guiding masking branches. The proposed method was applied to object detection task that captures both spatially and channel-wise features as the knowledge to supervise the student's learning (Yang et al. 2024a). A more recent study on logit-based KD with a hierarchical distillation mechanism was conducted by Xie et al. (2024). An integration of both logit and feature was utilised to mitigate the capacity gap between teacher and student models. In addition, the method employed feature matching and logit separation (Xie et al. 2024).

In the action recognition task, a generative model introducing feature-based knowledge and an attention-based mechanism was developed by Wang et al (2024) to improve the performance of small models. (Wang et al. 2024b). Feature-based KD can also play an important role in the recommender system, and Zhu and Zhang (2024) proposed a method called FreqD to focus on important knowledge by redistributing knowledge weights (Zhu and Zhang 2024). Shao et al. (2023) studied the Adversary-based Ensemble Feature Knowledge Distillation (AEFKD) technique. It allowed students to learn probabilistic information and high-dimensional features. It also identified feature map distributions in a model. (Shao et al. 2023). Yang et al. (2023) utilised a feature-based KD and attention mechanism to transform intermediate features to supervise the student model. They also highlighted the importance of

transferring knowledge that is related to reasoning (Yang et al. 2023). Yuan et al. (2024) used Semantic Graph Mapping (SGM) to transfer intermediate knowledge between different models at different scales, which overcame the limitations of traditional KD techniques (Yuan et al. 2024). Furthermore, Yang et al. (2024) integrated Vision Transformer (ViT) and KD, which identified the importance of both shallow and deep layers in ViT for distillation (Yang et al. 2024b). Feature-based KD has drawbacks such as challenges of matching features between different models (Wang et al. 2022b) and additional computation that is time-consuming (Phuong and Lampert 2019).

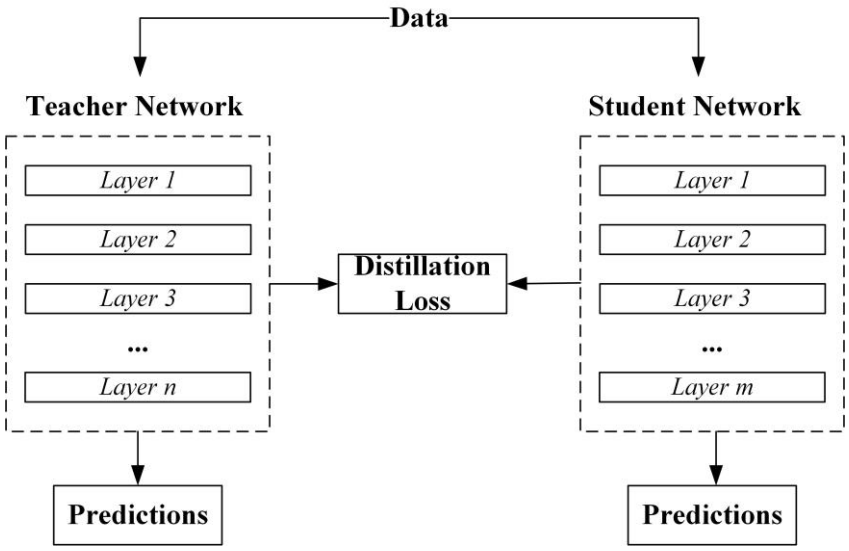


Figure 2.11 Feature-based KD architecture.

- ***Relation-based Knowledge Distillation***

Relation-based KD (Figure 2.12) focuses on the structural information through the output of a model, which is derived from the different layers or data samples. It allows students to learn the structural information of the teacher model (Gou et al. 2021).

To effectively train students with the relation-based KD, Yim et al. (2017) utilised the Gramian matrix to capture the relationships between two layers. This matrix described the relationships between two feature maps by calculating the products of features

from the two layers (Yim et al. 2017). Furthermore, Zhang et al. (2024) leveraged a token-level relationship graph to improve KD performance to facilitate the knowledge transfer from the teacher to student models, particularly in the context of dealing with unbalanced data (Zhang et al. 2024). Park et al. (2019) presented an approach for relation-based KD on instance relations (Park et al. 2019). According to the approach proposed by Passalis et al. in 2021, they transferred probabilistic knowledge to enhance KD performance (Passalis et al. 2021). By using contrastive learning, Tian et al. (2020) introduced contrastive representation distillation, which assisted student models in learning more knowledge from teacher models (Tian et al. 2019b). In the context of more recent research, Xin et al. (2024) utilised neighbourhood feature relationships and logit relationships for the distillation process to develop a KD-based similarity relationship. This approach transferred neighbourhood relation knowledge by selecting K nearest neighbours of each sample, and ultimately the knowledge was used to train the student model (Xin et al. 2024). To adjust the output queries of the student and teacher models, Li et al. (2024) proposed a KD scheme based on semantic segmentation and querying of instances of the transformers (Li et al. 2024).

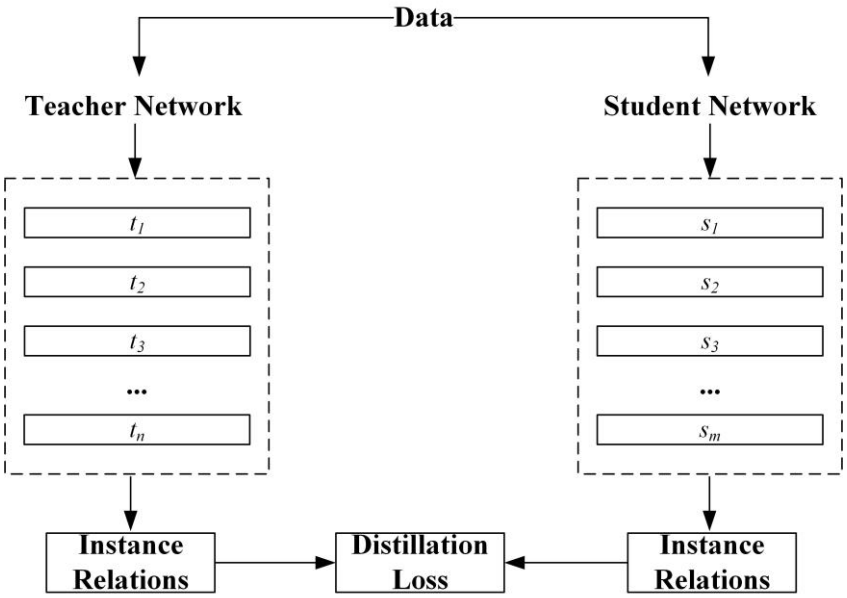


Figure 2.12 Relation-based KD architecture.

2.6 Field-Programmable Gate Arrays

The section will introduce the basics of Field-Programmable Gate Arrays (FPGAs) in Section 2.6.1, their use in CNN accelerators in Section 2.6.2, and their implementation in smart manufacturing in Section 2.6.3.

2.6.1 Fundamentals of FPGAs

FPGAs, a type of Integrated Circuit (IC), can be reprogrammed to implement various algorithms according to specific tasks. Modern FPGAs can be configured to implement a broad array of software algorithms, with the capacity for millions of logic cells (Guzel Aydin and Bilge 2021). While FPGA design processes resemble those of processors more than traditional processors, FPGAs offer significant advantages and often match or exceed their performance. A further advantage of FPGAs over ICs is their ability to be dynamically reconfigured (Rodriguez-Andina et al. 2007). The process is similar to loading a program into a CPU processor, but some or all of the resources available in an FPGA may be affected by this process. An FPGA architecture is composed of Look-Up Tables (LUTs), Flip-Flops (FFs), multiplexers, wires, and Inputs / Outputs (I/O) blocks (Boutros and Betz 2021). A significant part of its flexibility can be attributed to its essential component, the Configuration Logic Block (CLB) (Figure 2.13). CLBs are responsible for providing logic and storage. There are various blocks within the FPGA structure, including configuration logic blocks and others (Gandhare and Karthikeyan 2019), where the LUTs and FFs belong to the essential elements.

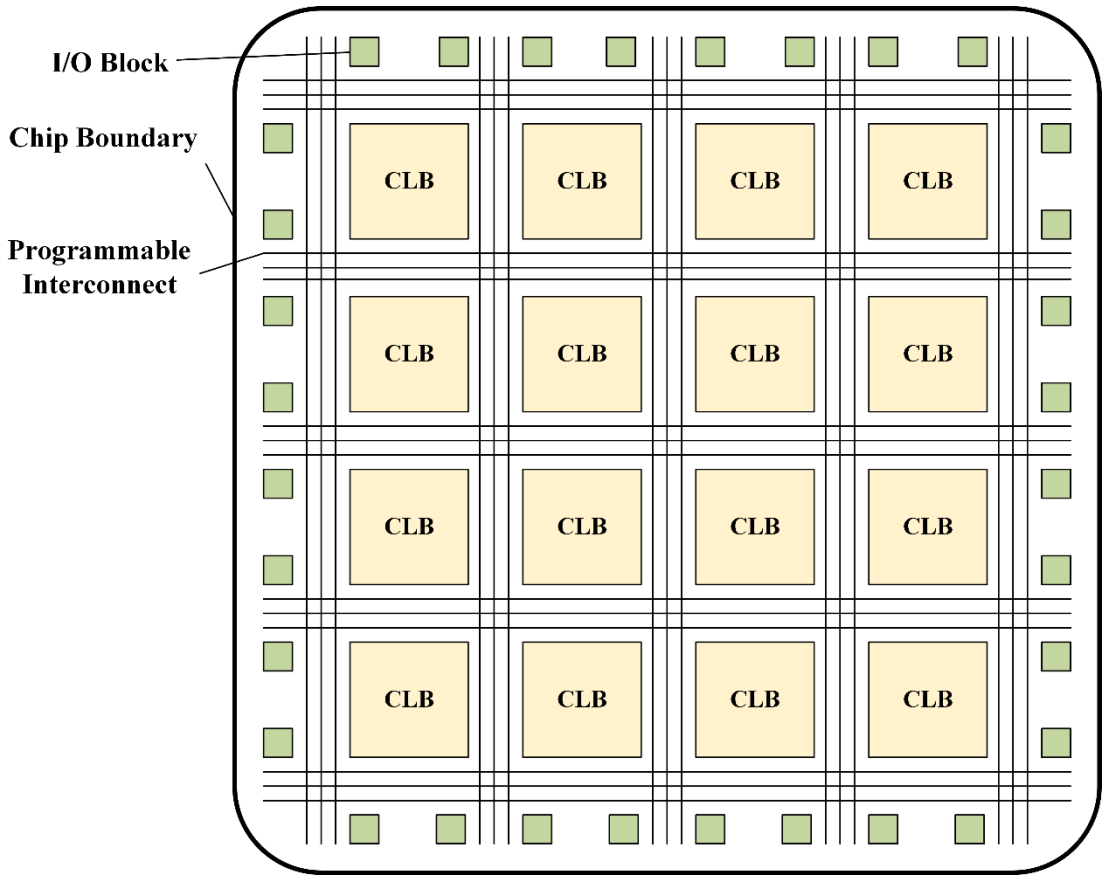


Figure 2.13 Detailed FPGA architecture including programmable logic and I/O.

In the case of a LUT, all binary-valued functions can be implemented with four binary inputs. Depending on the configuration of the LUT, the outputs enter a buffer block configured as a register (i.e. FF) (Mueller et al. 2009). FFs are the primary storage units in the FPGA architecture. As described above, the unit is always used with a LUT to aid in logic pipeline operations and data storage. Some other memory blocks can be utilised, such as Random-Access Memory (RAM), Read-Only Memory (ROM), and shift register (Rodriguez-Andina et al. 2007). Analogue circuitry is incorporated in blocks such as I/O and RAM (Boutros and Betz 2021). Each logic block contains several bits of memory and one or more LUTs. As a result, logic blocks can implement arbitrary logic functions. Programmable wires connect logic blocks into circuits of arbitrary complexity by routing logic blocks' outputs to each other's inputs (Draper et al. 2003).

FPGAs have become popular for DL applications due to their high computational processing, low power consumption and ability to adapt to different networks (Seng et al. 2021). According to the specific applications, FPGA has the potential to bridge the gap between Application Specific Integrated Circuits (ASICs) and embedded processors (Magyari and Chen 2022).

2.6.2 Accelerating CNNs with FPGAs

Graphics Processors (GPUs) and Central Processing Units (CPUs) have been utilised in the training and inference phases of DNN applications due to their computational power. However, DL models fail to be deployed directly, due to the limited on-chip resources of FPGAs (Schmitt et al. 2020). Considering latency and data volume challenges, it is crucial to reduce response time and computational burden. Various accelerators such as FPGAs, GPUs or even ASICs can improve the efficiency of CNN architectures. FPGA-based accelerators show the merits of good performance, high energy efficiency, a fast development cycle, and high reconfigurability (Guzel Aydin and Bilge 2021).

Research advances focused on using FPGAs to accelerate CNNs, which has several advantages over conventional microprocessors and ASICs. The unique characteristics of FPGAs contribute to high efficiency in parallel processing (Zhang et al. 2015), energy efficiency (Qiao et al. 2017), flexibility and reconfigurability (Mittal 2020), improved latency (Venieris and Bouganis 2017) and customisation for specific tasks (Guo et al. 2016). Abdelouahab et al. (2018) reviewed comprehensively accelerating CNN inference by FPGAs. This review indicated that convolution layers generate most of the workload, whereas fully connected layers generate most of the weights so that large models cannot be run in real-time. Mode compression techniques, such as weight pruning and low-rank approximations, play an important role in mitigating these limitations. By taking advantage of the sparsity of pruned CNNs in FPGA implementations, they suggested skipping the multiplication of zero weights for each layer during unrolling (Abdelouahab et al. 2018). Further development on CNN

accelerators was carried out by Philip and Sivamangai (2022), who showed prevailing optimisations and acceleration for CNN implementation on FPGAs, considering DL algorithms and FPGA design. In addition, FPGAs have been challenged in developing computational elements and control units within the current framework (Philip and Sivamangai 2022). Kumar and Madhumati (2023) employed FPGA-accelerated CNN model computation utilising multiple approximate accumulation units based on fixed-point data types. They employed the LeNet-5 accelerated network structure, which they validated on handwritten digits from the MNIST dataset (Kumar and Madhumati 2023). Various convolutional layers in the binary model were implemented using Finite State Machines (FSMs). According to the study by Pérez and Figueroa in 2021, a hardware accelerator based on MobileNet V2 inference was proposed for real-time image classification. This accelerator utilised loop tiling, bank-balanced pruning, dynamic quantisation, and off-chip storage for increased performance and reduced power consumption (Pérez and Figueroa 2021). Wu et al. (2023) redesigned processing elements implementation to share multiple functions and hybrid memory in order to maximise resource utilisation and achieve efficient inference (Wu et al. 2023b).

2.6.3 Smart Manufacturing Applications of FPGAs

FPGAs have the potential for real-time processing on the edge in the AM systems. It can accelerate computationally intensive tasks, such as DL models on the FPGA, conserving performance (Xu et al. 2022).

Many studies have focused on FPGAs to accelerate the model in different applications. Luo and Chen (2021) leveraged FPGAs to accelerate the defect detection process in AM. FPGA and DL were integrated to provide highly accurate and real-time defect detection. According to their findings, the targeted FPGA platform improved the efficiency and speed of defect detection in AM systems (Luo and Chen 2021). Scharf et al. (2019) introduced an FPGA-based vision system for in-line monitoring. This system was associated with image processing, contributing to the nominal operating and thereby enhancing AM process control and monitoring (Scharf et al. 2019).

Researchers such as Renken et al. (2019) also discovered the usage of an FPGA-based control system with FPGAs. The system consisted of different sensors to measure melt pool temperature in real-time and adjust the laser power accordingly to stabilise the temperature. By implementing the closed-loop and feedforward control strategies, temperature deviations were mitigated, resulting in up to a 90% reduction in temperature variance. The effectiveness was demonstrated in several geometries and conditions, including the bridge structures and powder-filled plates (Renken et al. 2019). An FPGA-based adaptive control system was developed by Rodriguez-Araujo et al. (2012) for industrial laser cladding processes, which improved monitoring and control over conventional PC-based systems. The system utilised FPGA technology to perform real-time image processing and control tasks in complex geometries and varying operating conditions. An adaptive control mechanism based on fuzzy rules dynamically adjusted control parameters to ensure a high-quality, consistent outcome enabled laser cladding variability and precision challenges to be addressed effectively (Rodriguez-Araujo et al. 2012). Ji et al. (2022) presented a model compression technique using KD and parameter quantisation to deploy DL models for bearing fault diagnosis on resource-constrained platforms like FPGAs, highlighting the practical applicability in real-world industrial settings (Ji et al. 2022).

By using real-time hyperspectral data processing integrated into hardware-in-the-loop, Devesse et al. (2016) developed a high-order physical model to improve precision in AM and adjust laser power based on real-time temperature feedback from the melt pool. This advanced approach was intended to improve the quality of manufactured parts (Devesse et al. 2016). A wire-based directed energy deposition process was also studied to address the challenge of maintaining constant layer heights. With a coherent range-resolved interferometric sensor, Qin et al. (2023) provided insights into accurate and efficient in-process geometric measurements of different process parameters, especially in the transition region. This FPGA-based signal processing system allows demodulation of the returned light directed to the photodetector (Qin et al. 2023).

To sum up, the FPGA platform is effective in many applications, including defect detection, real-time image processing, surveillance, and industrial control. In addition to these applications, it is notable that FPGA platforms have the potential to predict energy consumption in AM systems.

2.7 Summary

In this chapter, the current AM technologies, especially SLS were reviewed. After that, DL-based and data-driven approaches to AM were discussed, followed by model compression techniques, especially KD. This model compression technique can be employed to develop lightweight models. The DL models become lightweight and deployable on small devices such as FPGAs. A review of the research advances in FPGAs was presented at the end of this chapter. The following chapter will present an overview of the proposed framework for predictive modelling and energy consumption optimisation in an SLS system.

Chapter 3 A Framework for Predictive Modelling and Energy Consumption Optimisation in SLS

3.1 Introduction

The emergence of AM has become a promising manufacturing paradigm, by decentralising production geographically and bringing it closer to end customers. In addition, AM facilitates free-form product design, significantly contributing to sustainable manufacturing practices (Khorram Niaki and Nonino 2017). AM reduces additional tooling, material waste and resource usage, which potentially boosts an energy-efficient and sustainable production process (Majeed et al. 2021). Sustainability could increase the long-term vision and efficiency of the production life cycle. Additionally, it would enable manufacturers to make smart decisions and reduce material waste and energy management (Dunaway et al. 2017). Developing energy prediction models contributes to more comprehensive analytics to assist energy management. The highly precise models are anticipated to minimise costs and improve overall process sustainability (Kellens et al. 2017). This chapter will outline the comprehensive framework for energy predictive modelling and the optimisation of design-relevant parameters as well as energy consumption.

3.2 Research Framework for Energy Consumption Prediction in SLS

Traditional empirical models and physics-based models often fail to capture the complex insights and non-linear relationships between geometric information and energy consumption collected from the dynamic environment in SLS, since they heavily depend on feature engineering from high-dimensional data and non-linear mappings (Wang et al. 2024c). Recent data-driven approaches, DL, offer promising

alternatives to improve the accuracy of energy consumption prediction. These approaches can model complex patterns and discover relationships in SLS processes (Qin et al. 2018). However, the current constraints of DL-based models are often associated with significant computational overhead (Cheng et al. 2017). The increased computational requirements for inference have posed challenges for edge device implementation and deployment (Chen et al. 2021). The high parameter count of deep Convolutional Neural Networks (CNNs) limits the usage of FPGAs in terms of processing on the edge. In order to overcome these challenges, a comprehensive framework for energy consumption prediction needs to be developed. Figure 3.1 demonstrates the research framework for an energy consumption prediction and management system, covering three main topics: 1) data-driven energy prediction model including stages 1 and 2, 2) FPGA and lightweight model optimisation including stages 3 and 4, and 3) energy optimisation support through design-relevant parameter adjustment. Each stage is organised as follows:

In the **data collection and knowledge acquisition stage** in Section 3.2.1, the image data of unique layers sliced from CAD models, energy-relevant data collected from a power meter, as well as design-relevant parameters from part designs and process planning are obtained. **Image-based predictive modelling** in Section 3.2.2 describes how the multi-scale feature fusion model can process the layer-wise image data, known as the teacher model when employing KD techniques. Upon completing the teacher model training, **model compression** in Section 3.2.3 describes the utilisation of KD techniques. **Lightweight model deployment** in Section 3.2.4 describes the deployment of the lightweight model. After training this lightweight model, the quantisation technique is employed to further compress the parameters of the student model to deploy on the targeted FPGA platform, followed by redesigning and configuring to the FPGA. Subsequently, the deployment and acceleration of the student model are accomplished. In Section 3.2.5, the final stage of the study includes an **optimisation approach.** These features and insights are integrated into another DNN model, integrating with design-relevant parameters. Using the Particle Swarm Optimisation (PSO) algorithm can identify the best parameter combinations and achieve optimal energy consumption.

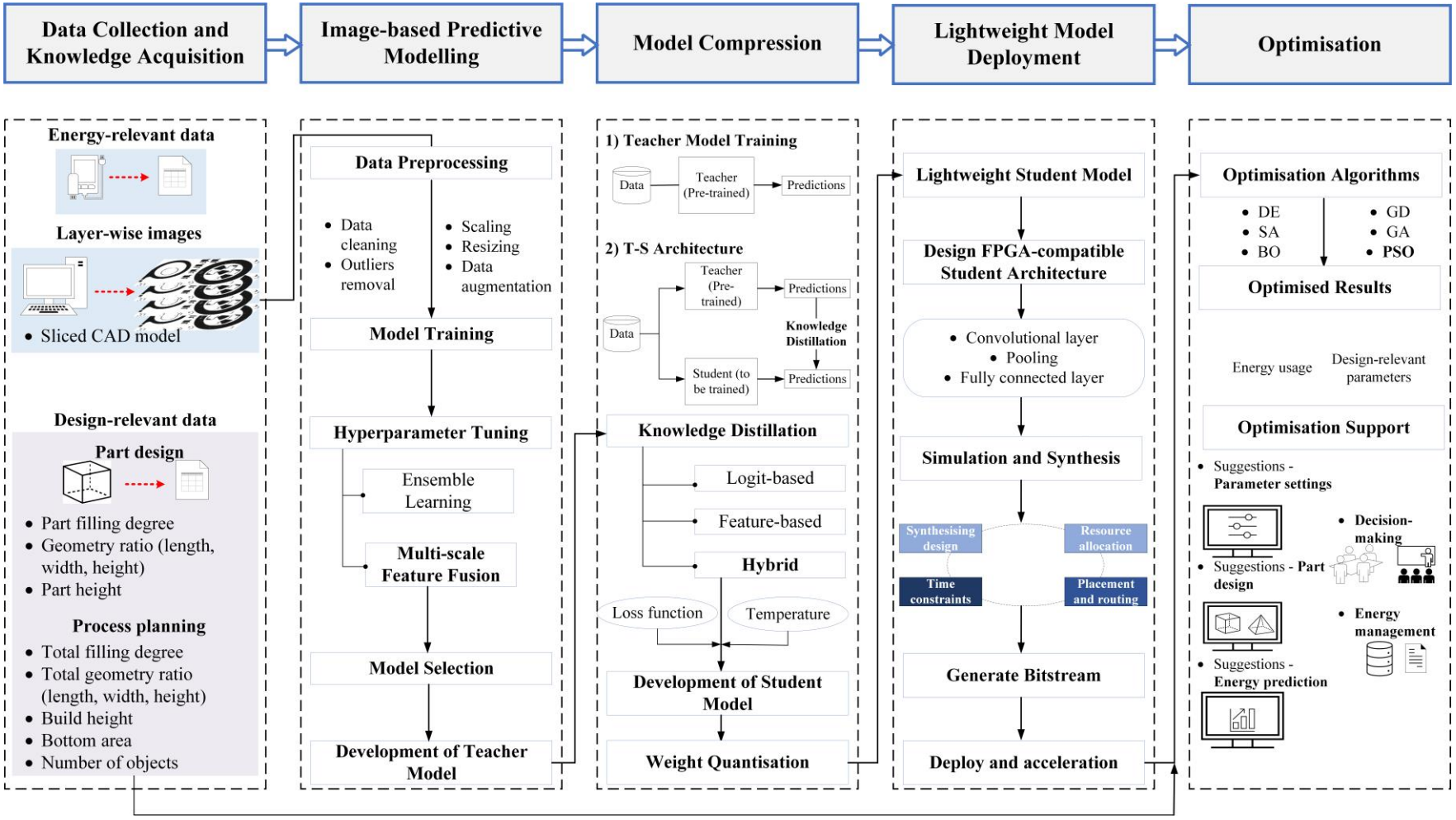


Figure 3.1 Framework for predictive modelling and energy consumption optimisation in SLS.

3.2.1 Data Collection and Knowledge Acquisition

Figure 3.2 illustrates the data obtained from the targeted SLS system. The dataset consists of three types of data including 1) layer-wise image data from sliced CAD models, 2) corresponding energy consumption on each unique layer, and 3) design-relevant data to be optimised, collected from another set of printed prototypes. The data is categorised into three distinct types based on their sources: energy-relevant data, layer-wise image data, and design-relevant data, each presenting different data types and levels of detail. Specifically, energy-relevant data are collected from a power meter, and layer-wise images are derived from distinct layers of the sliced CAD model. These datasets are utilised to train the multi-scale feature fusion model, which can extract the image features with the most significant impact on the energy consumption of each layer. Additionally, design-relevant data were gathered from part design and process planning before fabrication, which is a critical aspect of the overall process. However, these data are not included in the energy predictive model. Instead, they play a significant role in the optimisation approach to determine the minimum energy consumption for the selected builds.

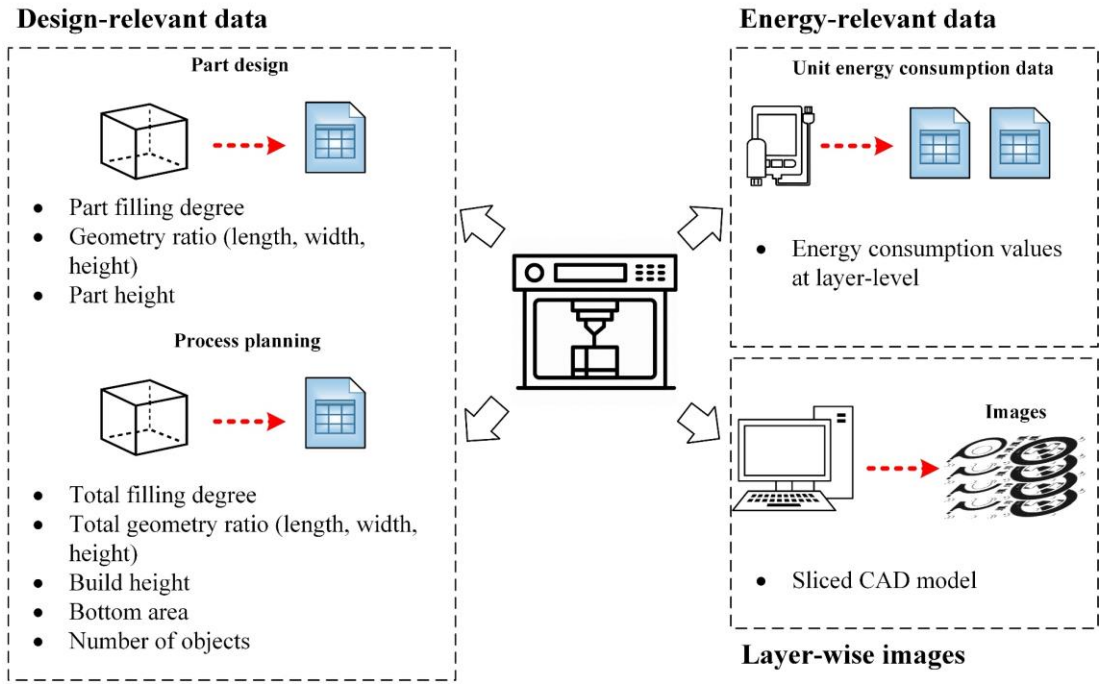


Figure 3.2 Categorisation of data collected in the SLS process.

- ***Data for Multi-scale Feature Fusion***

The dataset for DL-based modelling consists of layer-wise image data and corresponding energy consumption values. The image data come from the sliced files of 32 distinct CAD models, accounting for over 20,000 layered images. These images contain geometric information extracted from the CAD models. The energy consumption data are the unit energy consumption values for each unique layer, measured by a power meter, with the unit energy consumption per layer varying from 4 to 200 Wh/g. In the case study, the image data are the model input, and the labels are the unit energy consumption of each layer. The historical data were gathered from the EOS P700 SLS machine with PA2200 nylon powder to assess its performance and energy consumption. Using Autodesk Netfabb AM analysis software, image data can be sliced and extracted from the CAD models of various printed prototypes. Figure 3.3 illustrates the integration of 3D models and layer-wise images, derived from CAD models containing geometric information, in a real-world SLS scenario. In the case of building on CAD layers upon layers, CNN input was related to images on each layer, whereas the label data represented the corresponding layer-wise unit energy consumption. These images were inputted to the energy consumption for feature fusion and predicting the unit energy consumption.

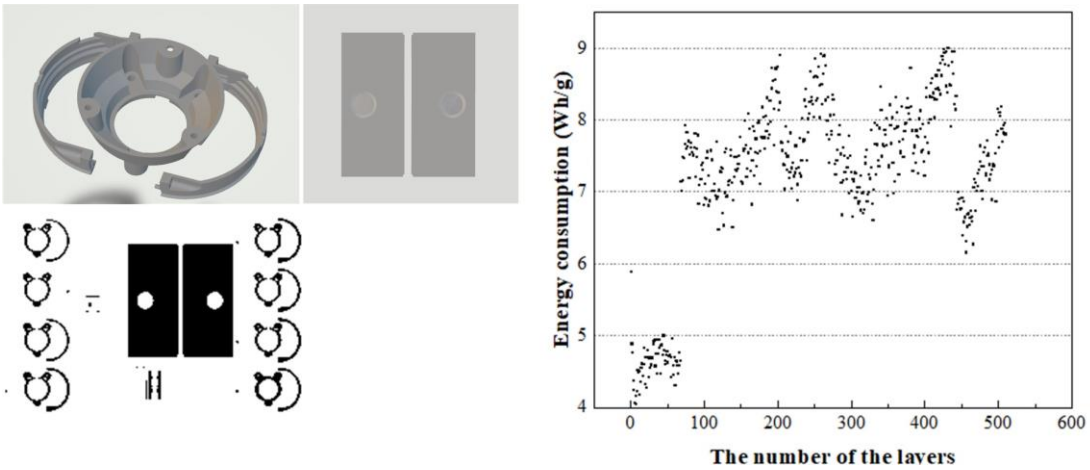


Figure 3.3 The sliced image data of a sample and the distribution of unit energy consumption (Wh/g) of one product.

In the preprocessing stage, both image data and corresponding energy consumption data will be processed. The first consideration is to observe and pre-process the collected image data. The removal of outliers within the energy-relevant data, such as extremely high and low energy consumption readings attributed to machine warm-up and cool-down, respectively, is applied. As a result of observing the input image dataset, it is necessary to remove some blank images, since they do not contain any useful information. This situation usually occurs in the first few images. Furthermore, the image datasets need to be resized and augmented to enhance the performance of the energy predictive model. In detail, the 128×128 pixels are extracted from the central region of the input images to minimise the impact on the significant features of the layer-wise images. These image data are also converted to grayscale images as the colour information does not influence the level of energy consumption. After completing the data preprocessing steps, the final datasets are integrated by combining energy-relevant data with image data. In this dataset, the energy-relevant data acts as the label for each corresponding image, which in turn represents a unique energy consumption value for a specific layer. In addition, when visualising the distribution of energy consumption, some anomalous data is always observed at the beginning of the data collection process, which is related to the pre-heating process. This data should be removed to avoid negatively affecting the performance of the model. After removing these blank images and the corresponding energy consumption at the preheating stages of the 3D printer, the Interquartile range (IQR) method is applied to detect and process the outliers again.

- ***Data for Optimisation Support***

Figure 3.4 illustrates the samples of the CAD design model to which the optimisation algorithms and DL were applied, thereby minimising the build energy consumption through an optimised combination of parameters.

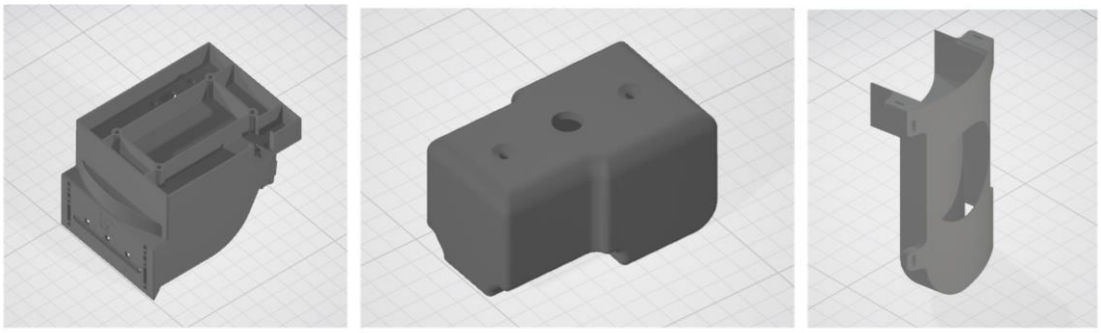


Figure 3.4 The samples of build for optimisation.

Table 3.1 and Table 3.2 summarise values of design-relevant parameters of three selected builds for optimisation, categorised into part-design and process planning parameters. This dataset with design-relevant parameters was defined by part designers and process operators. These parameters can be categorised into part-design and process planning parameters. Specifically, part-design parameters were established by designers according to the unique design requirements, whereas process-planning parameters were set by operators before the process, focusing on design and layout considerations. These parameters include aspects such as rotation and positioning, filling degree, part height and the quantity of parts.

Table 3.1 The descriptions of part-design data.

Name of Parameters	Descriptions	Build 1	Build 2	Build 3
Degree of Part Filling (%)	The proportion of the actual volume versus the total volume of a single part	12.03	17.59	23.10
Part Ratio (WL) (%)	The proportion of the length and width of a single part	1.06	1.31	1.23

Name of Parameters	Descriptions	Build 1	Build 2	Build 3
Part Ratio (HL) (%)	The proportion of the length and height of the single part	0.61	1.68	0.44
Part Ratio (WH) (%)	The proportion of the height and width of a single part	1.73	0.78	2.80
Part Height (mm)	The height of the part	106.51	188.5	40.83

Table 3.2 The descriptions of process-planning data.

Name of Parameters	Descriptions	Build 1	Build 2	Build 3
Degree of Total Filling (%)	The proportion of the actual volume versus the total volume of the build	11.17	9.35	4.69
Total Ratio (WL) (%)	The proportion of length and width of the entire build	0.55	0.53	0.49
Total Ratio (HL) (%)	The proportion of the length and height of the entire build	0.54	0.82	0.17
Total Ratio (WH) (%)	The proportion of the height and width of the entire build	1.02	0.64	2.87
Bottom Area (cm^2)	The bottom area	2585.51	2546.88	1917.71
Height (mm)	The height of the build	371.02	570.68	107
Num of Part	The number of build	24	54	10

3.2.2 Image-based Predictive Modelling

At the beginning of this stage, conventional modelling techniques are utilised, with a focus on CNNs. Subsequently, a series of image-based predictive models are integrated into an ensemble model to enhance the predictive performance for energy consumption from layer-wise images. Different models will be compared to identify the most effective and efficient performance.

However, the traditional models fail to perform multi-scale feature fusion image data, particularly in predicting energy consumption by using layer-wise image data. There are two approaches to developing the multi-scale feature fusion model. As an initial attempt, a Feature Pyramid Network (FPN) constructs a hierarchical feature pyramid with features at multi-resolutions by exploiting contextual information at different scales across the entire network in a top-down manner (Lin et al. 2017). Considering the energy consumption influenced by layer-wise images, the Spatial Pyramid Pooling (SPP) module is adopted to enhance the capability to extract contextual information (He et al. 2014). For the energy consumption prediction in SLS, the multi-scale nature of part geometry will have an influence, so that SSP can generate features by pooling operations from the global perspective, which is key to understanding the complexity of the energy demand for the entire components.

The U-Net architecture is introduced. U-Net combines features from image data at different scales and captures image information at diverse levels (Ronneberger et al. 2015). In the SLS scenario, the encode-decoder architecture of U-Net avoids manual design on feature fusion. By using the skip connection mechanism, the high-resolution features can directly be transferred to the decoder, enhancing modelling by considering more local design features of layer-wise images such as holes and edges, which is essential to the energy consumption prediction. In addition, this architecture is suitable in scenarios with high annotation costs since it requires less annotated data. The

proposed architecture incorporates two modules: the Asymmetric Convolution (AC) block (Ding et al. 2019) and the Convolutional Block Attention Module (CBAM) (Woo et al. 2018). This architecture effectively combines multi-scale contextual information. The skip connections in U-Net facilitate the integration of high-resolution features from earlier layers with low-resolution features from deeper layers. This improves the ability to capture details while maintaining a broader context preserves spatial information and improves localisation accuracy.

3.2.3 Model Compression

Existing DL-based models are limited by expensive computation and intensive memory requirements. It makes them unsuitable to be directly deployed on devices with limited memory resources or in applications demanding low latency. Model compression strategies utilising Knowledge Distillation (KD) and quantisation techniques are necessary for developing efficient and effective lightweight models for edge platforms.

According to KD, the behaviour of a large teacher network trains and supervises the learning of a student model. This process effectively transfers knowledge while maintaining a lightweight structure through the proposed KD strategy. In the proposed methodology, the work focuses on exploring different KD strategies such as logit-based KD, feature-based KD and hybrid KD with logit and attention-based knowledge. In detail, logit-based knowledge is regarded as the output of the teacher model i.e., prediction of energy values, which helps understand the relationship between input images and corresponding energy consumption. Feature-based knowledge represents the features at the intermediate layer of the teacher model, such as the edge or shapes of the layer-wise images. Relation-based knowledge is often the interaction between images or the combination of image features that lead to significant changes in energy consumption. Combining the scenario, there is no need for utilising relation-based knowledge to guide the student model. Firstly, considering energy consumption prediction, predicting energy consumption from layer-wise images is the priority rather

than the relationship between two successive layer-wise images. Secondly, the dataset does not include temporal information to support learning. Compared to it, logit and feature-based knowledge are more intuitive to this specific task. The work is finally established on the logit and feature-based distillation, guiding the student model to concentrate more on the response of the teacher model and the specific area of the image feature. Leveraging the attention mechanism can also improve the learning capability of the student model, especially in terms of intermediate interpretability and details on the sliced model, which could influence energy consumption.

Parameter quantisation can reduce the number of bits required to represent each parameter, significantly reducing model complexity and computational demand (Choudhary et al. 2020). This method will be employed to further minimise the memory requirements and model complexity of the predictive model, making it suitable for the targeted FPGA platform. Combining KD and quantisation techniques enables the creation of lightweight and high-performing models, essential for deployment in environments with limited computational resources.

3.2.4 Lightweight Model Deployment

Before deploying on the targeted FPGA platform, the parameters of the student model are quantised to fit within the limited resources available in the FPGA. The quantisation technique plays a crucial role in reducing model complexity by approximating the representation of a DL model that uses floating-point numbers with one that uses low-bit width numbers. Quantisation results in the requirement for fewer bits to represent weights and biases in a CNN. These quantised parameters prevent storage and memory shortages and reduce computational complexity. In addition to parameter quantisation, the architecture of the student model needs to be redesigned to accommodate the on-chip resources and logic on the targeted FPGA platform. Specifically, this design includes optimising the algorithm, adjusting the data flow process and ensuring the compatibility between the student model and the targeted platform.

The student model can achieve faster prediction to obtain features and energy consumption on the targeted FPGA platform. Subsequently, these features will be integrated into a DNN, which utilises the PSO technique to optimise design-relevant parameters and minimise the energy consumption of printed prototypes.

3.2.5 Optimisation

The integration of PSO and DL performs as an optimiser, which optimises a set of design-relevant parameters to minimise the energy consumption of a prototype. These parameters incorporate design-relevant data on the level of the entire build, such as filling degree, part rotation and position, bottom areas, and total height. PSO plays a key role in iteratively providing optimal parameters to minimise energy consumption, thus evaluating the impact of these parameters on energy consumption. Upon reaching the minimum energy consumption values, the algorithm produces the optimal parameters corresponding to that minimum value. These optimal parameters are invaluable to part designers and process operators for establishing the most effective geometry and other key design selections, such as the optimal part ratio between length and width or part filling degree before the process.

FPGA-based CNNs significantly enhance predictive analytics from image data derived from CAD models for AM energy consumption. During the offline training phase, predictive analysis is employed before the manufacturing process. The deep feature fusion architecture is implemented to predict and mitigate potential energy consumption increases. This pre-trained model transfers knowledge to the lightweight student model. Once applied to the FPGA platform, it accelerates the extraction of valuable information and predicts layer energy consumption from the lightweight model, integrating with design-relevant parameters such as part design and process planning. This setup allows the AM system to conduct swift layer-wise image processing, significantly speeding up the extraction of hidden and significant features by the CNN. This predictive model provides valuable insights from historical data,

optimising parameters across different designs, thereby identifying the lowest energy consumption based on these parameter combinations.

This framework can contribute to an informed decision-making process in the design by providing these optimal parameter combinations to part designers and process operators. Based on outcomes from the energy prediction model, designers and operators can leverage these insights to create more energy-efficient designs in SLS. The framework integrates prediction and optimisation capabilities to enhance the energy efficiency of the targeted SLS. Furthermore, it demonstrates a data-driven approach to industrial energy management in SLS systems.

3.3 Summary

This chapter outlined the framework for developing energy consumption and management in an SLS system. The framework includes data collection and knowledge acquisition, predictive modelling and feature fusion, model compression and lightweight DL development, model deployment, and optimisation in terms of energy and design-relevant parameters. In the following Chapter 4 to Chapter 6, the detailed research methodology will be demonstrated, followed by significant findings and insights that can contribute to this proposed framework.

Chapter 4 Data-driven Modelling for Energy Consumption Prediction with Knowledge Distillation in SLS

4.1 Introduction

In the past decades, AM has shown its merits of reducing material and resource utilisation as well as tooling requirements. It has demonstrated the potential for energy conservation and sustainable production (Majeed et al. 2021). Advances in data sensing and collection technologies increase data availability in AM processes. The current development of more comprehensive energy modelling and management strategies is suitable for the dynamic working environment. With an ability to process substantial data volume and computational power, DL can process and discover valuable energy-relevant information and insights from data, which is critical to effective decision-making and optimisation. On the other hand, its modelling time is long and computationally expensive. Therefore, it is crucial to balance model compression and performance in fixed architecture models, which makes techniques such as KD essential. The research leverages the KD technique in balancing model performance and model complexity for edge deployment in AM energy management.

This chapter is the preliminary step of the proposed framework. It involves applying the ensemble approach and logit-based KD techniques to develop the lightweight student model for predicting energy consumption by using layer-wise image data, laying the foundation for subsequent FPGA deployment and parameter optimisation.

4.2 Overview of Knowledge Distillation-based Predictive Modelling

The proposed methodology employed a teacher-assistant model between teacher and student models. This KD initialisation is divided into three components: teacher ensemble, teacher assistant, and a distilled student model through KD, followed by a subsequent validation phase. To begin this process, a teacher ensemble was established to enhance the algorithmic performance of individual teachers. The teacher ensemble was trained independently by comprising three different CNNs. To bridge the capacity gap between the pre-trained teacher ensemble and student model, a teacher assistant model was employed, which could improve the generalisation ability of the student model. During the distillation process, the two student models that were not fine-tuned were learned from the teacher assistant, which was used to compare the performance of the proposed methodology. Following this, a case study based on an SLS system was conducted to demonstrate the feasibility and effectiveness of energy consumption prediction modelling.

Figure 4.1 demonstrates the energy consumption prediction framework based on logit-based KD and a teacher assistant model. The first step involved training multiple CNNs using layer-wise images and energy-relevant data to develop a baseline model. The second step developed an ensemble teacher model to improve the performance of individual CNNs. Subsequently, a teacher-assistant network was developed between the teacher ensemble and the student to bridge the capacity gap between models. Finally, the KD process was employed to transfer knowledge to shallow CNNs, which were then used to predict energy consumption.

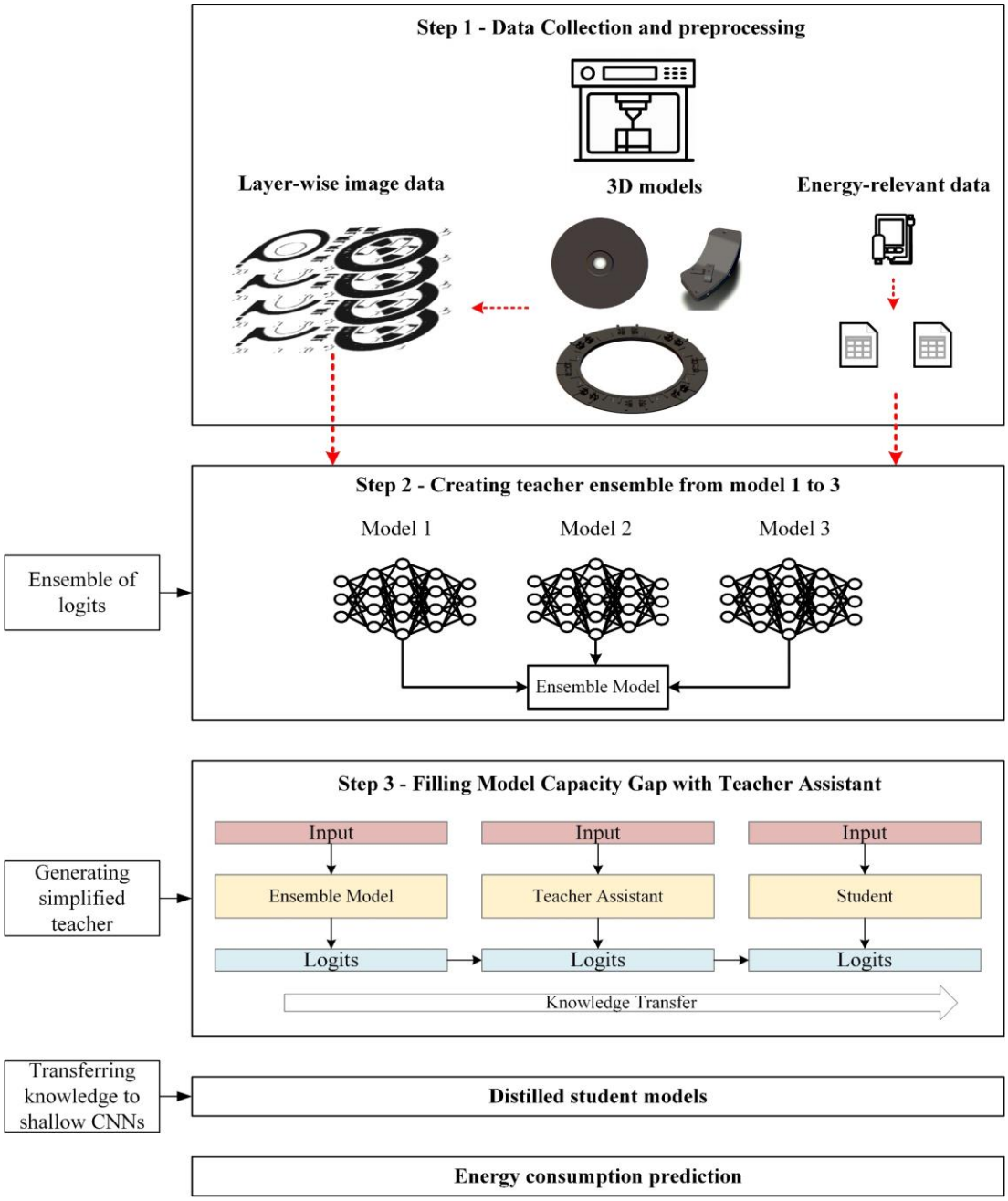


Figure 4.1 Workflow diagram of the proposed methodology.

4.2.1 The Strategy of Ensemble Approach

The task is associated with feature extraction from the layer-wise images, in which the image-based prediction model will be employed. For example, the architecture of CNNs is dedicated to processing image data and learning directly from the structural data without human intervention. It typically accepts input with three dimensions:

height and width of images and colour channels in RGB, while the colour channels can be neglected since they have less impact on energy consumption. Hence, the teacher leverages the single-channel image data (i.e., grayscale images) as the model input. However, a single model has less learning capability to perform image feature extraction, requiring a more robust and effective approach to the task.

Superior predictive performance can be achieved by employing an ensemble of multiple learning algorithms rather than relying on any single learning algorithm alone. The primary objective of using an ensemble is to identify a hypothesis that may not exist within the hypothesis space of the individual models that constitute it. Ensembles tend to yield better empirical findings when there is significant variation among the models (Jing Yang et al. 2013). To effectively reduce variance, an ensemble strategy for Deep Neural Networks (DNNs) is proposed. Three different learning algorithms showed relatively poor prediction performance in the experiments. An ensemble can achieve improved performance by using diverse DL algorithms or varying hyperparameters. To enhance the overall performance of the teacher network and achieve diversity among different base learners, three different models were integrated into the experiment (El-Rashidy et al. 2020). Furthermore, incorporating more weak learners can negatively impact training and inference times. The stacking approach involves training a model to combine other models (models 1, 2 and 3 in this case study). The ensemble architecture is depicted in Figure 4.2.

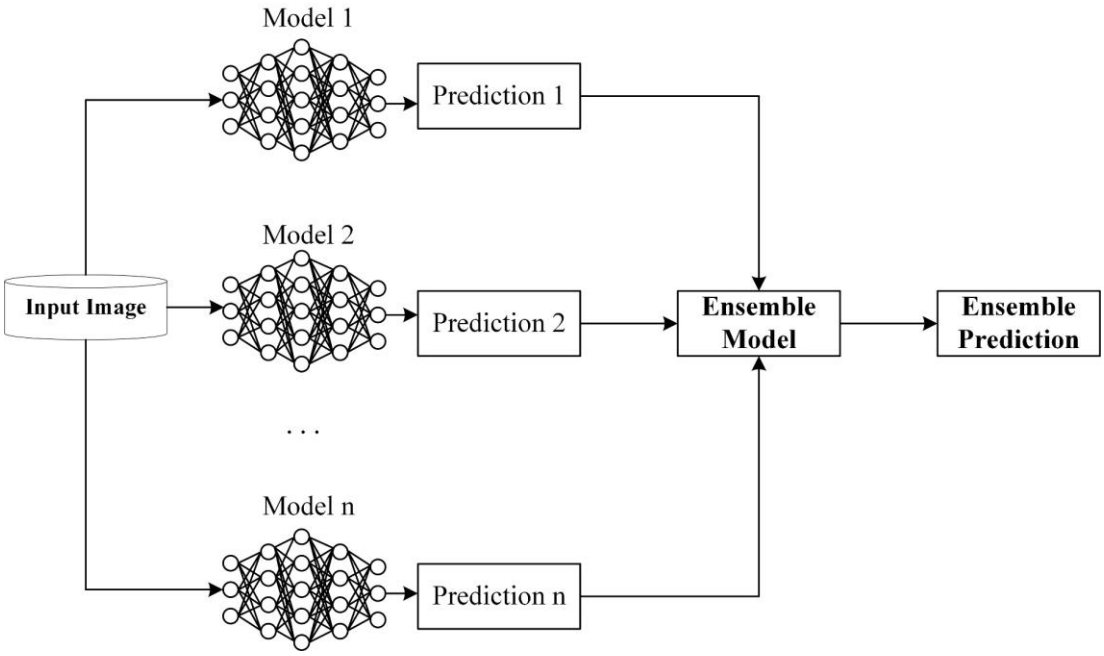


Figure 4.2 Ensemble method in ML.

The stacking approach involves training a combination of different models to integrate predictions from various learning algorithms. This integration is achieved by averaging the outputs of the models. An ensemble stacks multiple models, requiring that input data for each model experiences forward propagation. Consequently, computations become more time-consuming, and training times are extended. A unified soft label, derived from the outputs of multiple teacher networks, is used to guide the training of student models (Fukuda et al. 2017). Based on Equation (4.1), if $\boldsymbol{h}$ represents the soft label of the teacher, $\boldsymbol{T}$ is the total number of the models, and $\boldsymbol{w}$ is a set of weights, then the overall result can be expressed as:

$$h(x) = \frac{1}{T}\sum_{i=1}^{T} w_i h_i(x)$$
$$where, w_i \geq 0, \sum_{i=1}^{T} w_i = 1 \quad (4.1)$$

The advantage of combining multiple models lies in their ability to reduce similar errors on the test set by averaging out errors from different models (Goodfellow et al. 2016). According to published results, combining multiple models can balance bias and variance. This results in predictions that are less sensitive to specific training data, training strategies, or the randomness of individual training runs.

Three different CNNs are selected and trained individually on the image dataset. Their performance is then evaluated on a test set. Following that, the ensemble model performance is evaluated through the integration of the three models. The ensemble is expected to outperform each model on the test set. Consequently, each teacher model analyses the data and generates predictions. The predictions are aggregated. Using these data and soft labels, the model is trained to achieve consistent performance across all teacher models. A distilled student model can be developed using labelled data for fine-tuning.

4.2.2 Logit-based KD

KD commonly typically employs a teacher-student architecture, where the large model serves as the teacher and the small model as the student. A simplified model may struggle with complex problems, and the training data may not be fully generalised. The teacher model's knowledge can teach the student model how to generalise beyond the training data with additional available predictions. Additionally, soft labels can provide more information than hard labels, indicating the degree of similarity between classes. The loss function is defined in Equation (4.2) below:

$$L_{distill} = -T^2 \sum_{i}^{N} q_i^T \log(p_i^T) \tag{4.2}$$

Here, $\boldsymbol{q_i^T}$ represents the logits generated by the teacher, $\boldsymbol{p_i^T}$ denotes the logits of the student model, and $\boldsymbol{T}$ is a hyperparameter named temperature that controls the smoothness of the probability distribution. According to Equation (4.3), the loss $\boldsymbol{L_{student}}$ is the student's total loss, which is the sum of loss of soft label loss $\boldsymbol{L_{soft}}$ and hard label $\boldsymbol{L_{hard}}$, which is balanced by hyperparameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, where $\boldsymbol{\beta} = (\boldsymbol{1} - \boldsymbol{\alpha})$. During distillation, the objective function comprises a distillation loss. $\boldsymbol{L_{distill}}$, corresponding to the soft target and a student loss corresponding to the hard target $\boldsymbol{L_{hard}}$.

$$L_{student} = \alpha L_{soft} + \beta L_{hard} \tag{4.3}$$

Equation (4.4) illustrates that the softened class probability distribution of the student model denoted as $\boldsymbol{P_i^T}$, which is influenced by the similarity of the student to the teacher. The key to KD lies in the design of the loss function, which includes common cross-entropy losses: $\boldsymbol{L_{soft}}$ and $\boldsymbol{L_{hard}}$, both based on the soft target. By incorporating the hyperparameter $\boldsymbol{T}$, soft loss $\boldsymbol{L_{soft}}$ quantifies the discrepancy between student model output and labels, using cross-entropy loss. Hard loss $\boldsymbol{L_{hard}}$ is the critical distillation loss, which measures the difference between the output of the student model and the output of the trained teacher model after distilling.

$$L_{soft=} \sum_{i}^{N} q_i^T \log(p_i^T) \tag{4.4}$$

The training data is input into both the teacher and student models, with the soft target being the softmax distribution generated by the teacher model.

4.2.3 KD with a Teacher Assistant

The idea of employing a teacher assistant stems from multi-phase distillation approaches, as demonstrated in the studies from Mirzadeh et al. (2020), where self-distillation has improved the accuracy of the base model (Mirzadeh et al. 2020). There is a potential for a model capacity gap when comparing a large DNN to the smaller one utilised by students, which can degrade the performance of the knowledge transfer. To facilitate effective knowledge transfer to student models, the introduction of a teacher assistant is suggested to manage the complexity of the model effectively. This approach involves reducing the structural differences between the student and the teacher ensemble, with network pruning and KD, directly compressing the energy consumption prediction model. The ensemble model trains students by using teacher assistants, rather than relying on a single, large teacher. The teacher ensemble may not possess the same level of expressiveness as students, so the teacher assistant is a smaller model with the same architecture. The teacher assistant can transfer teacher predictions that the student might otherwise fail to express. The teacher assistant is positioned at the intermediate level between the teacher ensemble and the student model. In comparison to the previous teacher ensemble model, the teacher assistant is engaged to match its performance while enhancing learning abilities. It is important to focus on softer targets to prevent any weakening of knowledge transfer from the teacher model to the student model.

4.3 Experimental Design and Setups

Employing multiple teacher models provides the student model with a diverse range of knowledge, potentially more beneficial than learning from a single teacher. The knowledge of large models, such as DNNS or ensembles of numerous models, is typically greater than that of small models, but this potential is sometimes underutilised. To assign probability distributions to numerous labels in DNNs during prediction tasks, a softmax layer is employed, outputting probabilities that reflect correlations among each class. However, the model assigns a lower probability to incorrect labels compared to the correct ones, leading to hard labels that do not account

for their prior knowledge. Students and teachers are given training data, with the student model generating a soft target based on the softmax distribution of the teacher model. This loss function includes the output of the student model's softmax and the cross-entropy of the soft target at the same temperature. The multi-stage KD consists of three stages: 1) establishing the ensemble of teachers, 2) establishing the teacher assistant model, and 3) establishing the student model. Stage one involved training the teacher ensemble while ensuring soft labels are preserved. Model capacity gaps were reduced by a teacher assistant in stage two, which addressed structural differences between models.

4.3.1 Experiment Setups

Data preprocessing was performed before the experiment. All images were collected and resized to 128×128 pixels and converted to grayscale, which subsequently constituted the final dataset, with 70% allocated to the training set, 20% to the testing set and 10% to the validation set. The predictive modelling, based on the teacher-student architecture, included a teacher ensemble, a teacher assistant, and a student model. The teacher ensemble was trained with three different CNN models to achieve optimal performance. All models utilised the same dataset to train the energy consumption prediction model. To validate the effectiveness and feasibility of the proposed KD-based approach, there were four groups of the experiment: 1) training each CNN individually with image datasets, 2) training the teacher ensemble, 3) training within the teacher-student architecture without a teacher assistant, and 4) training with the proposed approach. In Experiment 3 and Experiment 4, the other two CNNs served as student models for comparative analysis.

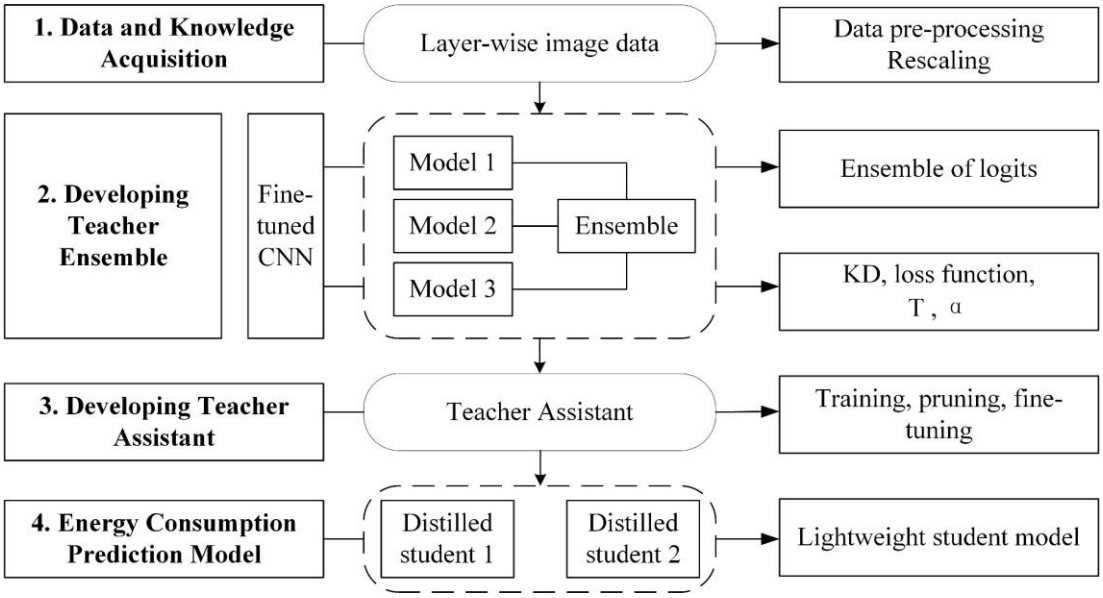


Figure 4.3 Workflow of model training and implementation for the proposed approach.

Firstly, three CNN models were directly trained on the AM dataset to establish the baseline model. The role of CNNs was to extract features from layer-wise image data derived from CAD models. The unit energy consumption of each unique layer can be predicted more accurately when the model is more complex and extensive, leading to higher computational costs. Thus, a balance between model complexity and computational cost is essential. The number of parameters in each CNN was different. Model 1, the simplest, contained approximately 4.43 million parameters. With additional high-level convolutional layers, Model 2 and Model 3 had more than 8.76 million and 10.68 million parameters, respectively. The three models were employed to train the ensemble model, aiming to achieve better results, but the ensemble could lead to a significant computational burden during training. The number of models in the ensemble was determined by the integrated model performance, typically an odd number was preferred (Liu et al. 2016; Troussas et al. 2020). Furthermore, the training time increased with the number of base models. Thus, three base models were selected for the ensemble.

Secondly, averaging combined the outputs of the three models in the top layer of the ensemble. The ensemble model had more parameters approximately 13.01 million, making it significantly more complex than the three individual models. Using hold-out validation data, these weights could be optimised based on the predictions from each model, thereby slightly enhancing the ensemble performance. In the context of the teacher ensemble, a student could not learn complete knowledge from multiple teachers, leading to a decrease in model accuracy.

Thirdly, the pruning technique reduced the parameter counts of the teacher ensemble while minimising accuracy losses. It aimed to eliminate a greater of the less relevant and redundant parameters. Weights are effectively set to zero through pruning based on their magnitude. By removing weights near zero or with low magnitude, the impact on the network is minimised. In addition, the model ensemble initiates with 50% sparsity (50% zeros in weight) and concludes with 80% sparsity, fine-tuning the pre-trained teacher ensemble from the previous step. This experiment aimed to determine whether the model complexity affected the student's final predictions. If so, an early-stopping mechanism was implemented to mitigate this phenomenon.

The final step utilised a teacher assistant model to mitigate the difference between teacher and student models, which was a pruned version of the teacher ensemble. The key to distillation relied on the loss function. The student output employed a softmax function to match the output of hard labels from the teacher model until the teacher model completed training. The temperature was incorporated into the softmax classifier of the completed teacher network, serving as a fitting target for the student network with the same temperature. Hyperparameter $\boldsymbol{\alpha}$ was introduced to determine the total loss function followed by the final training. It is recommended to set $\boldsymbol{\alpha}$ to 0.9 and $\boldsymbol{T}$ within the range of {3,4,5} (Huang and Wang 2017). A distiller was constructed with the following configurations: a pre-trained teacher ensemble model, a student model for training, and a student loss function ($\boldsymbol{\alpha}$ set to 0.9 and $\boldsymbol{T}$ to 3).

4.3.2 Evaluation Metrics

AM poses the challenge of consuming significant processing time, which contributes to increased energy consumption. The overall energy utilisation is highly dependent on the duration of the manufacturing process, as longer processing times typically result in higher energy demands. The energy consumption level can be evaluated by Specific Energy Consumption (SEC) (Wh/g) or unit energy consumption, denoted as $\boldsymbol{E_U}$, as shown in Equation (4.6), where $\boldsymbol{E_T}$ and $\boldsymbol{M_T}$ are total energy usage and mass of a total part, respectively. The energy consumption $\boldsymbol{E_U}$ is determined by the proportion of power rate and process productivity, reflecting the efficiency of the energy used in the manufacturing process.

$$E_U = \frac{E_T}{M_T} \tag{4.6}$$

The performance of the proposed method can be evaluated using three key metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Model Correlation Coefficient (MCC), which provide comprehensive insights into the accuracy and reliability of the model.

The evaluation metrics of the predictive model involve RMSE and MCC. As shown in Equation (4.7), RMSE indicates the difference between the actual value $\boldsymbol{a_t}$ and the predicted value $\boldsymbol{p_t}$, which the low RMSE results in the high accuracy of the model performance. In Equation (4.8) and Equation (4.9), MCC reveals the correlations between the predicted and actual data obtained from the model, where $\overline{\boldsymbol{p}}$ is the mean value of predicted data, and $\overline{\boldsymbol{a}}$ is the mean value of the entire data.

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(p_t - a_t)^2} \tag{4.7}$$

$$MCC = \frac{S_{PA}}{\sqrt{S_P S_A}} \tag{4.8}$$

$$S_{PA} = \frac{\sum_i (p_i - \bar{p})(a_i - \bar{a})}{n-1};\ S_P = \frac{\sum_i (p_i - \bar{p})^2}{n-1}\ ;\ S_A = \frac{\sum_i (a_i - \bar{a})^2}{n-1} \tag{4.9}$$

4.4 Results and Discussions

4.4.1 Results of Baseline Models Training

Figure 4.4 and Figure 4.5 illustrate the RMSE and MAE of three different mode CNN over 20 epochs. The RMSE and MAE of the teacher ensemble directly trained from image data in these two figures are significantly lower than others. At the start of training, the teacher ensemble has the second-lowest RMSE and MAE. These RMSE and MAE values decrease as the training. The ensemble approach thus improves energy consumption predictions and reduces errors as anticipated. The teacher assistant model is then pruned based on this teacher ensemble in the subsequent stage.

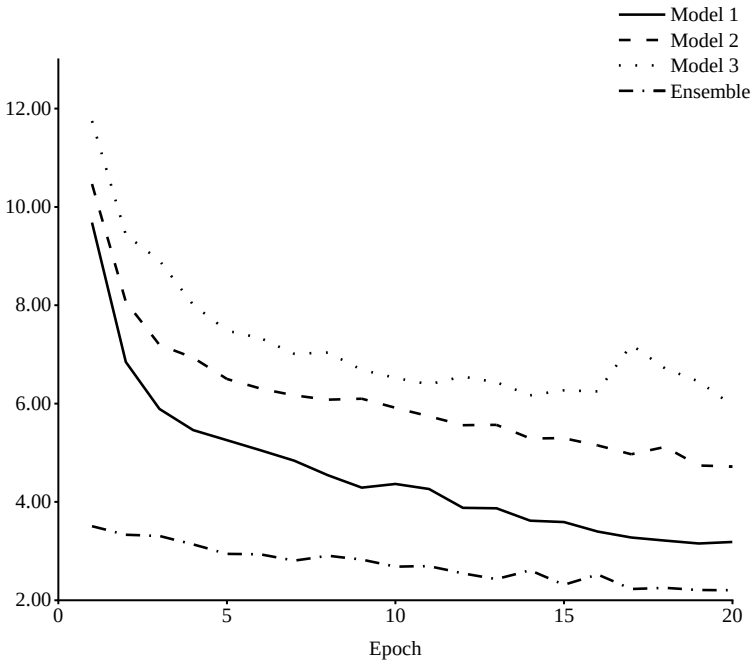


Figure 4.4 RMSE comparison of trained models in terms of CNNs and ensemble predictions.

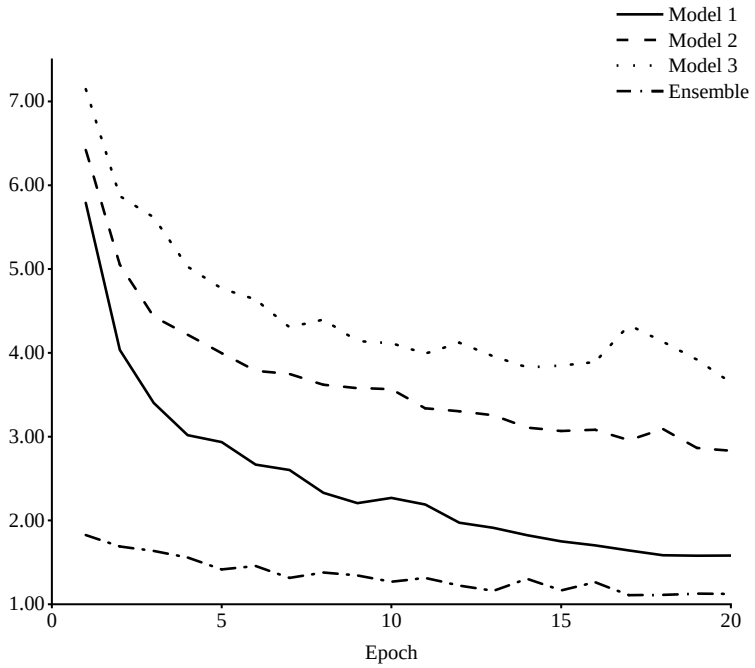


Figure 4.5 MAE comparison of trained models in terms of CNNs and ensemble predictions.

4.4.2 Results of KD with Teacher Assistant and Ensemble Teacher Model

Table 4.1 Comparative model performance in energy consumption prediction in baseline and ensemble model.

Model No.	RMSE (Wh/g)	MAE (Wh/g)
1	11.61	9.01
2	10.05	6.88
3	12.72	12.03
Ensemble	**9.94**	**6.72**

Table 4.1 highlights that the ensemble model outperforms with the lowest values of 9.94 Wh/g for RMSE and 6.72 Wh/g for MAE. Experimental results are detailed in the subsequent contents. These findings show that three different prediction models yield relatively poor prediction performance. To enhance the overall performance of the teacher network and to contribute to diversity among the weak learners, three different models were integrated into the ensemble.

Table 4.2 shows that the ensemble method leads to high computational demand. The KD technique effectively balances the trade-offs between performance and model complexity. KD-based approaches aim to reduce the model complexity while maintaining significant performance. Observing findings from the two experiments in Table 4.2, weak CNNs correspond to a reduction in model performance compared to the ensemble, thereby highlighting the advantage of ensemble architecture. The ensemble model typically outperforms any individual models, as it combines weak learners to reduce the variance, optimising the model output. Additionally, Table 4.2 demonstrates that adding high-performing ensemble members can mitigate errors and

optimise error rates. In terms of knowledge derived from diverse data in the AM dataset, multiple-teacher models offer superior guidance compared to a single-teacher model. The teacher network integrates and synthesises various knowledge representations from individual teacher models after an ensemble prediction. Larger models with more parameters, tend to perform better within the ensemble. Though more effective, the ensemble model demands more computation, leading to longer prediction times during deployment.

Table 4.2 Comparative model performance analysis: baseline, ensemble and teacher assistant models.

Experiment No.	Teacher Model	#Params	Model Size (MB)	Training Time per epoch (s)	RMSE (Wh/g)	MAE (Wh/g)
Experiment 1	1	4.42M	12.20	101	11.61	9.01
	2	8.77M	49.12	184	10.05	6.88
	3	10.68M	87.74	179	12.72	12.03
Experiment 2	Ensemble	**13.01M**	**149.06**	**259**	**9.94**	**6.72**
	TA	/	/	**90**	**18.14**	**11.08**

In Experiment 1, with an increase in the number of layers in the CNNs, the RMSE and MAE values for Model 1, Model 2 and Model 3 exhibit variations. Additionally, training these models on AM data demands a significant amount of training time. Model 3 requires 179 seconds per epoch, compared to Model 1 which requires 101 seconds. The ensemble model shows a high level of computational complexity due to the large number of parameters involved, as evidenced by its size of approximately 13.01 million parameters, which requires a longer training time of 259 seconds per epoch. This results in a longer training time of 259 seconds per epoch. In Experiment 2, the RMSE and MAE of the teacher ensemble are significantly lowered to 9.94 Wh/g

and 6.72 Wh/g, respectively. To improve the generalisation ability of the student model, a teacher assistant distils and simplifies information from the teacher ensemble. Compared to the ensemble model, the teacher assistant model features a marginally lower error rate. This network achieved faster training time and minimised the error between the student and the teacher ensemble through the pruning process. Increasing the number of layers in the final CNN also limits the applicability of the ensemble model in real-world deployment scenarios. Thus, a pruning approach is employed to compress the model complexity thereby reducing training time. This pruning strategy may negatively affect model performance, potentially impacting the final performance of the distilled models, despite being designed as a simplified version of a pre-trained teacher ensemble.

Table 4.3 Illustration of model performance in terms of distilled student models.

Experiment No.	Student Model	#Params	Model Size (MB)	Training Time (s)	RMSE (Wh/g)	MAE (Wh/g)
Experiment 3	Distilled student A	1.19M	1.94	95	9.14	5.48
	Distilled student B	2.20M	3.72	110	9.56	6.16
Experiment 4	Distilled student A	1.19M	1.94	**98**	**9.03**	**5.30**
	Distilled student B	2.20M	3.72	**106**	**7.39**	**3.77**

According to Table 4.3, Experiment 3 was conducted without incorporating a teacher assistant as an intermediate stage between the teacher ensemble and the students. As parameter counts increased, Student A and Student B in different architecture, which were not fine-tuned, outperformed the ensemble teacher model while maintaining

faster training time. Model size exceeds expectations when compared to models developed through independent training. Distilled student models offer advantages over independently developed ensemble models, particularly in terms of training time and model size. In Experiment 4, a teacher assistant model was employed. The experimental results presented in Table 4.3 indicate that single models exhibit significantly higher error rates compared to the ensemble. Complex CNNs are more effective at meeting the high predictive demands of image data.

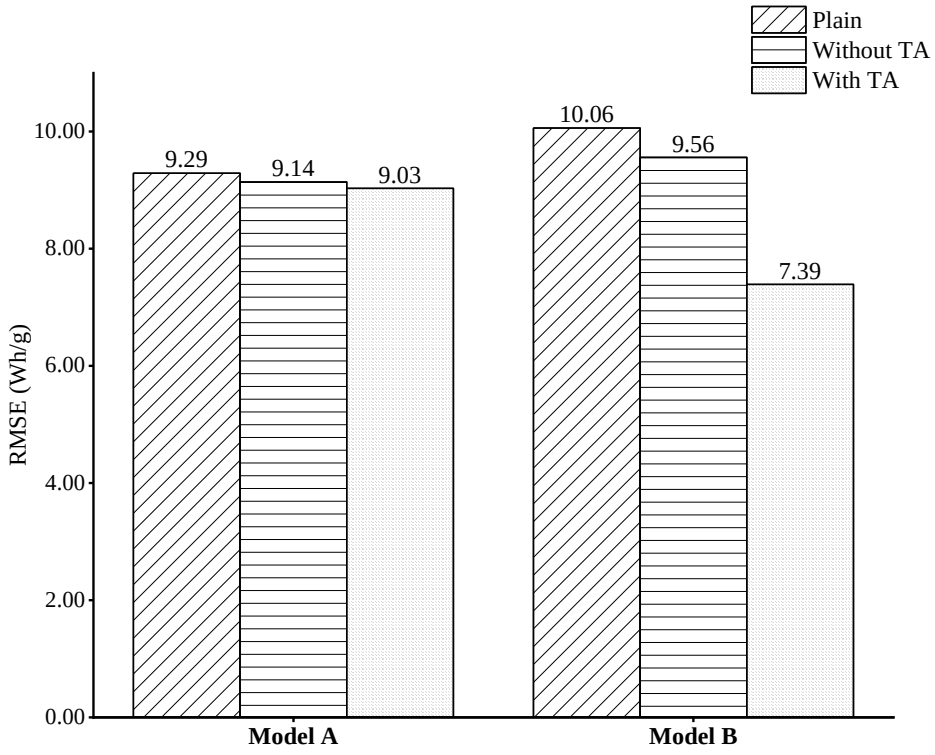


Figure 4.6 Comparative RMSE analysis of distilled student model A and B with Original model.

Figure 4.6 and Figure 4.7 illustrate an overall downward trend in the pattern. The RMSE for model A is 9.29 Wh/g, while the MAE for model B is 10.06 Wh/g. Following the initial training of the first two models, both the teacher assistant model and direct training employing KD were subsequently applied. The employment of TA to Model A results in a 2.8% reduction in its RMSE. A similar pattern can be observed in Model B with a significant reduction of 26.5% in RMSE values. The RMSE value for Model A decreases from 5.727 Wh/g to 5.3 Wh/g, while that of Model B decreases

from 6.91 Wh/g to 3.77 Wh/g, when employing the teacher assistant network. These findings indicate that the knowledge of the teacher model can effectively transfer to the student, which is more capable of generalisation than the original student model. Additionally, employing a teacher assistant model between teacher and student models could bridge the model capacity gap, which achieves a level of performance of both teacher and student models.

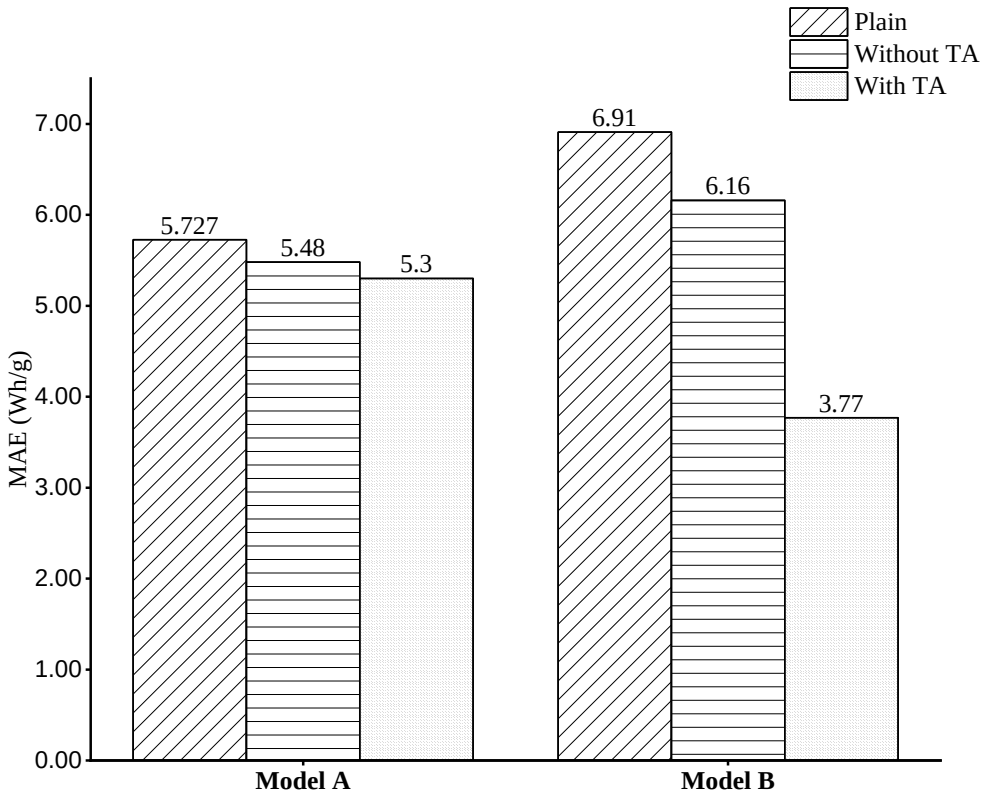


Figure 4.7 Comparative MAE analysis of distilled student model A and B with Original model.

An additional experiment has been conducted, incorporating process parameters within the model. Within the predictive model framework, these process parameters are integrated with part geometry characteristics including the maximum and minimum values for dispenser speed, recoater speed, hatch power, hatch speed, and hatch width. Table 4.4 summarises the main findings of introducing the process

parameters as the experimental data. The results indicate a marginal difference in predictions when comparing models that incorporate only part geometry versus those that combine part geometry with process parameters. This marginal difference may be attributed to the slight variations in process parameters across builds, as well as the relatively minor influence of these parameters on unit energy consumption.

Table 4.4 Comparative analysis of experimental results based on geometry features and process parameters.

	RMSE (Wh/g)		**MAE (Wh/g)**	
	Model A	Model B	Model A	Model B
Part geometry	9.03	7.39	5.30	3.77
Part geometry + process parameters	9.08	7.64	5.41	2.85

4.4.3 Results of Energy Consumption Prediction with Distilled Student Models by Using KD

The results depicted in Figure 4.8 and Figure 4.9 correspond to the two student models, Model A and Model B, after the KD process. Model A shows a mean unit energy consumption of 9.89 Wh/g, with values ranging from 2.64 Wh/g to 37.63 Wh/g and a standard deviation of 6.70 Wh/g. The actual unit energy consumption is 11.89 Wh/g, so the prediction of Model A is close to that of the actual value. On the other hand, Model B has a mean unit energy consumption of 16.66 Wh/g, ranging from 6.67 Wh/g to 77.74 Wh/g with a standard deviation of 11.61 Wh/g. This result indicates that the prediction of Model B has a greater variability. Both models provide reasonable accurate predictions of energy consumption based on the layer-wise image data after employing the proposed method. However, predictions in Model A are more consistent and closer to the actual energy consumption, making it more compatible and reliable for the practical energy predictions.

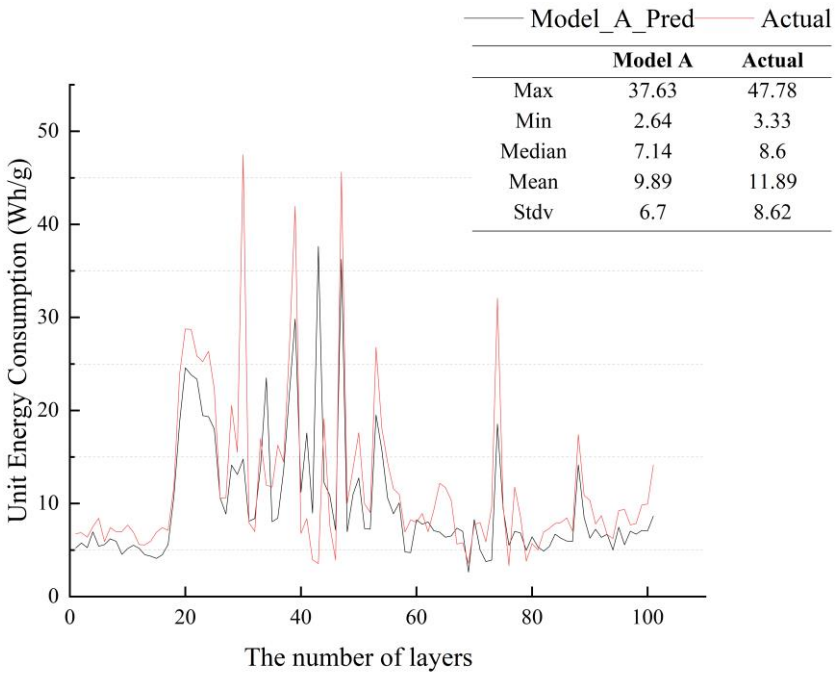


Figure 4.8 Unit energy consumption prediction accuracy for model A.

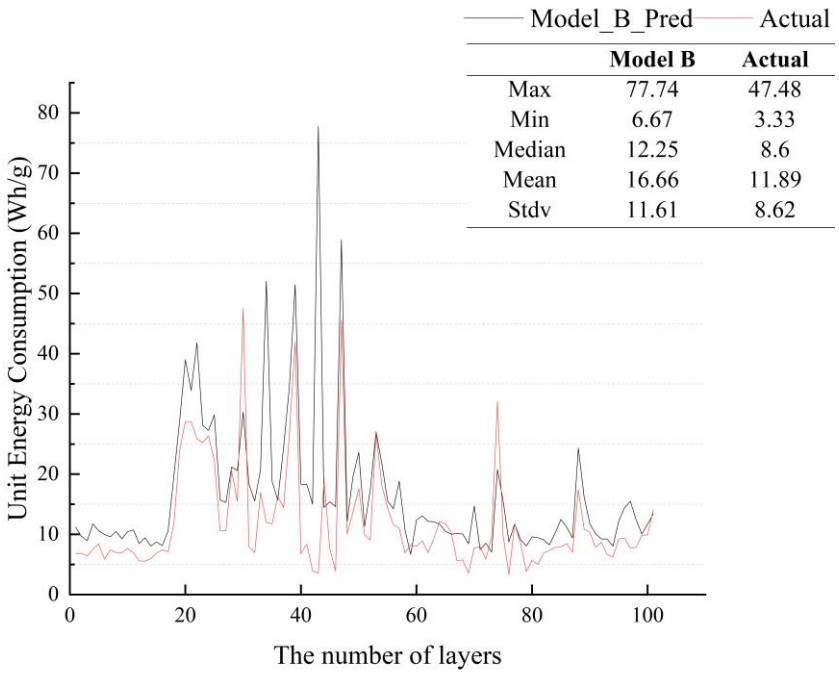


Figure 4.9 Unit energy consumption prediction accuracy for model B.

Improvement in the performance of the teacher ensemble enables it to provide more effective supervision to the student model, serving as a superior predictor. However, Since the architecture of the teacher ensemble becomes increasingly complex, the student model may not be able to fully learn its predictions due to the gap between model capacity. Therefore, a teacher assistant is employed to mitigate this model capacity, bridging the teacher ensemble and the student model, resulting in the desired performance. Utilising soft labels derived from the teacher ensemble, the TA plays a critical role in mitigating errors according to Experiments 3 and Experiment 4 in Table 4.3. A second observation from Figure 4.6 and Figure 4.7 is that the employment of the teacher assistant model enhances the outcomes in comparison to a distilled student model without a teacher assistant.

Despite the limitations arising from the reduced number of variables in the student mode, the soft labels provided by teachers are crucial in mitigating these issues. Employing student models on training sets yields comparatively high performance. Differently from other compression techniques, the proposed KD-based approach implements a multi-step framework that transfers critical knowledge from a complex teacher model to a simpler student model via a teacher assistant model, considering both model complexity and performance. As part of such a teacher and student architecture, the knowledge extracted from the teacher is transferred to the student model while minimising error. The distilled student model shows degraded performance due to the large structural difference between the teacher ensemble network and the student network.

4.5 Summary

This chapter introduces a data-driven approach for energy consumption prediction by using KD. It aims to establish a predictive model by using layer-wise images obtained from a targeted SLS system. Specifically, this approach leverages CNNs to extract valuable features that impact the energy consumption of the unique layer. These features are readily extracted using conventional data analytics. At this stage, the

results are deemed acceptable. KD aims to compress the model, making it suitable for the deployment environment. Consequently, the next stage offers substantial potential for advancing the optimisation of the proposed architecture that is designed for deployment and acceleration. This enables a specialised edge platform to be equipped with efficient DL models while minimally affecting performance. However, there is scope for further enhancement in energy prediction by image data and optimisation based on the algorithm and model architecture in the subsequent research since the ensemble approach has an oversimplified feature fusion capability, which is less efficient for more complex image features. Besides, TA with pruning technique lost critical parameters leading to the reduced performance of student models. In the next chapter, the research focuses on leveraging FPGAs and lightweight models after KD for predictive modelling in the targeted AM system.

Chapter 5 Leveraging FPGAs and Lightweight Neural Networks for Predictive Modelling in SLS

5.1 Introduction

This chapter explores predictive modelling using the lightweight model derived from the KD for energy consumption prediction. Furthermore, this chapter discusses the deployment of the lightweight model on FPGAs. The study of energy consumption prediction from design information in AM systems faces two main challenges: accurately capturing features from layer-wise images and efficiently integrating these features into predictive models at different levels. A multi-scale feature fusion model can address these challenges. It overcomes the limitations of traditional data-driven approaches for learning complex, design-relevant datasets acquired from the SLS system. Energy consumption prediction can be significantly improved by using multi-scale feature fusion techniques and continuous learning capability.

5.2 Integration of FPGAs and Lightweight NNs for Predictive Modelling in SLS

5.2.1 Multi-scale Feature Fusion for Energy Consumption Modelling

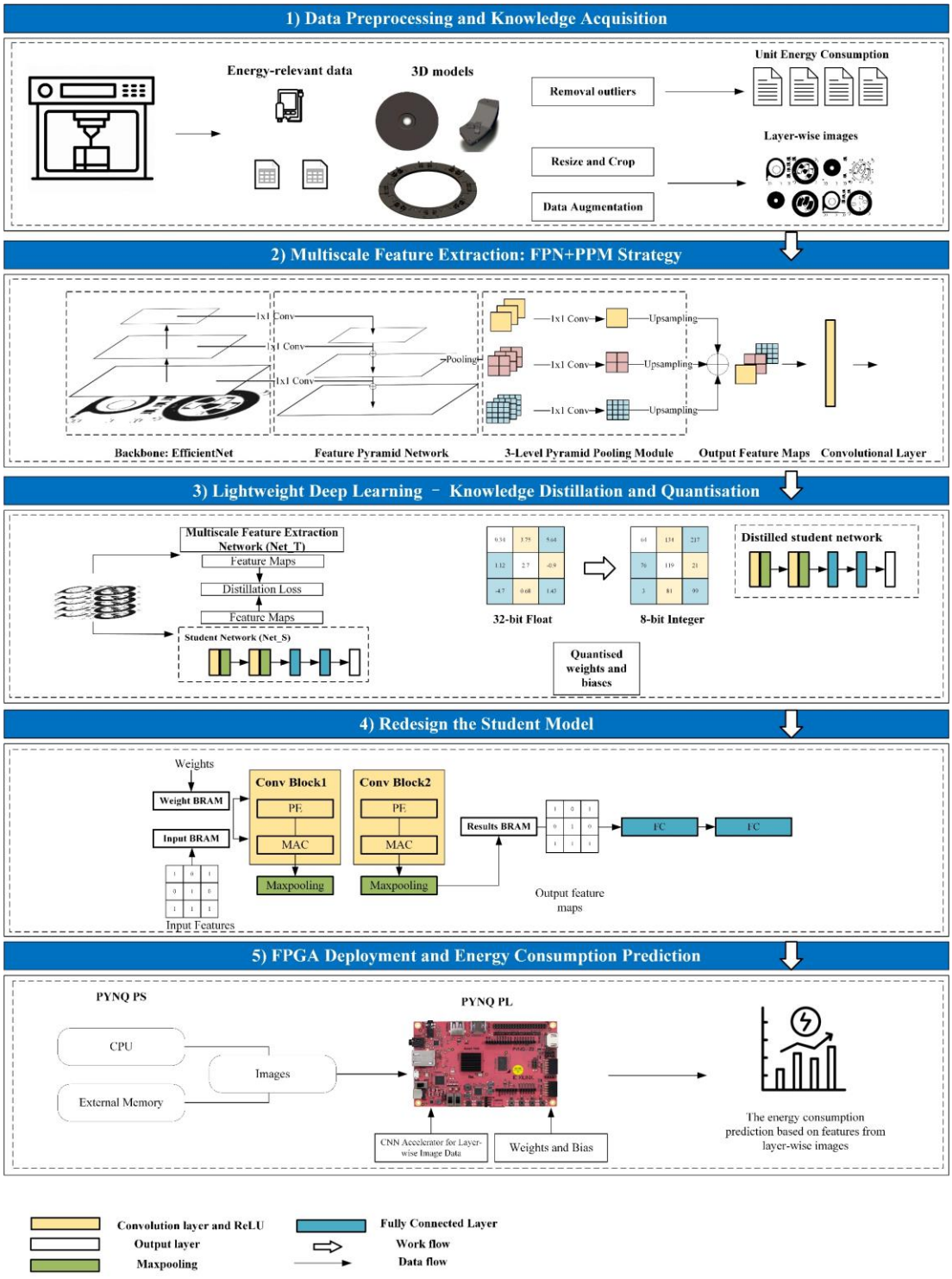


Figure 5.1 Detailed workflow of FPGA-CNN integration for predictive energy modelling.

Image data and corresponding energy consumption data were pre-processed for analysis. Specifically, the 128×128 pixels were extracted from the central region of the input images to minimise the impact on the significant features of the layer-wise images. It is essential to remove blank images from the image dataset since they lack useful information, which indicates the process is idle or warming up. In addition, it is observed that anomalous data are present after visualising the energy consumption distribution. The anomalous values in datasets represent the process in the preheating or cool-down process, which should be removed to prevent negative effects on the model performance. The interquartile range method was utilised to detect and process any remaining outliers in energy consumption.

The proposed architecture for energy consumption prediction leveraged the traditional Feature Pyramid Network (FPN) architecture (Lin et al. 2017), which combined the Spatial Pyramid Pooling (SPP) module (He et al. 2014) and EfficientNet as a backbone. Subsequently, a feature pyramid was utilised to extract the initial features of the model. The extracted feature maps were combined with 1×1, 2×2 and 4×4 pooling layers to integrate the feature maps across different layers. In addition, a 1×1 convolutional layer reduced channel numbers while preserving the critical feature information. Finally, each processed feature map was up-sampled to its original dimensions, and the up-sampled feature maps were concatenated with the original feature maps along the channel dimension to create the feature representation needed for training the student network.

5.2.2 Feature-based KD Strategy

According to Romero et al. (2015), feature-based knowledge was initially utilised to improve the training of student networks in the FitNet study (Romero et al. 2015). It is shown in Equation (5.1) that the distillation loss involves intermediate features, where $\boldsymbol{f_t(x)}$and $\boldsymbol{f_s(x)}$ are the intermediate features of teacher and student networks, respectively. When two networks do not have the same shape, two transformation functions $\boldsymbol{\Phi_t\big(f_t(x)\big)}$ and $\boldsymbol{\Phi_s\big(f_s(x)\big)}$ are employed. The similarity function $\boldsymbol{\mathcal{L}_F}$ is

determined by comparing the feature maps of teacher and student networks (Gou et al. 2021).

$$L_{FeaD}\left(f_t(x), f_s(x)\right) = \mathcal{L}_F\left(\Phi_t\left(f_t(x)\right), \Phi_s\left(f_s(x)\right)\right) \tag{5.1}$$

To accurately match teacher and student representations of features, further investigation is needed into the significant difference in size between the hint layer and the guiding layer (Gou et al. 2021). The feature-based KD technique is employed to enable student networks to learn knowledge from teacher networks. It focuses on knowledge transfer through the intermediate layers of both networks. However, the difference in feature scales between the two different networks leads to additional linear matching in the distillation process. It is difficult to distil feature information when teacher and student networks are not in the same architecture.

Smooth L1 loss reduces the sensitivity to outliers in the data due to its balance between Mean Squared Error (MSE) and Mean Absolute Error (MAE), as shown in Equation (5.2).

$$L_{SmoothL1}\left(f_t(x), f_s(x)\right) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}\phi\left(f_t(x), f_s(x)\right) \tag{5.2}$$

where $\boldsymbol{\phi}(\boldsymbol{x})$ is defined as:

$$\phi(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| < 1 \\ |x| - \frac{1}{2} & \text{otherwise.} \end{cases} \tag{5.3}$$

A loss value less than this parameter should be estimated using the MSE. Otherwise, the MAE is calculated. With larger loss values, the MAE mitigates the impact of outliers, resulting in a more balanced model. For smaller loss values, MSE is used to maintain a quadratic function near the centre.

The concatenated feature maps provided by the SPP module were used as features to train the student network. Notably, the Smooth L1 loss can be applied in the context of feature-based KD by considering both ground truth labels (task-specific loss $\boldsymbol{L_{task}}$) and the proposed architecture as a teacher network distillation loss ($\boldsymbol{L_{distill}}$), demonstrated in Equation (5.4) and Equation (5.5).

$$L_{task} = L_{smoothL1}(f_t(x), y) \tag{5.4}$$

$$L_{distill} = L_{smoothL1}\big(f_t(x), f_s(x)\big) \tag{5.5}$$

Equation (5.6) sums up the losses ($\boldsymbol{L_{total}}$), where $\alpha + \beta = 1$:

$$L_{total} = \alpha \cdot L_{task} + \beta \cdot L_{distill} \tag{5.6}$$

5.2.3 FPGA-based Prediction Model Implementation

- ***Student Model for Processing Image-based Data***

In the context of deploying on a compact computing platform, the student model depicted in Figure 5.2 exhibits a reduced scale in complexity compared to its teacher model. This architecture comprises two convolutional blocks and two fully connected

layers, accompanied by a single output layer. The student model demonstrates enhanced efficiency on smaller computing platforms due to its smaller size while preserving sufficient structural integrity to capture the hidden features of the image data. Employing two convolutional blocks and two fully connected layers, complemented by an output layer, allows the student model to concentrate on the most important features of the data and predict energy consumption. Consequently, the student model generates a reduced number of weights and biases within its convolutional and fully connected layers, compared to the teacher model. By employing the proposed feature-based KD technique, the student model may exhibit compromised generalisation ability in comparison to the teacher model.

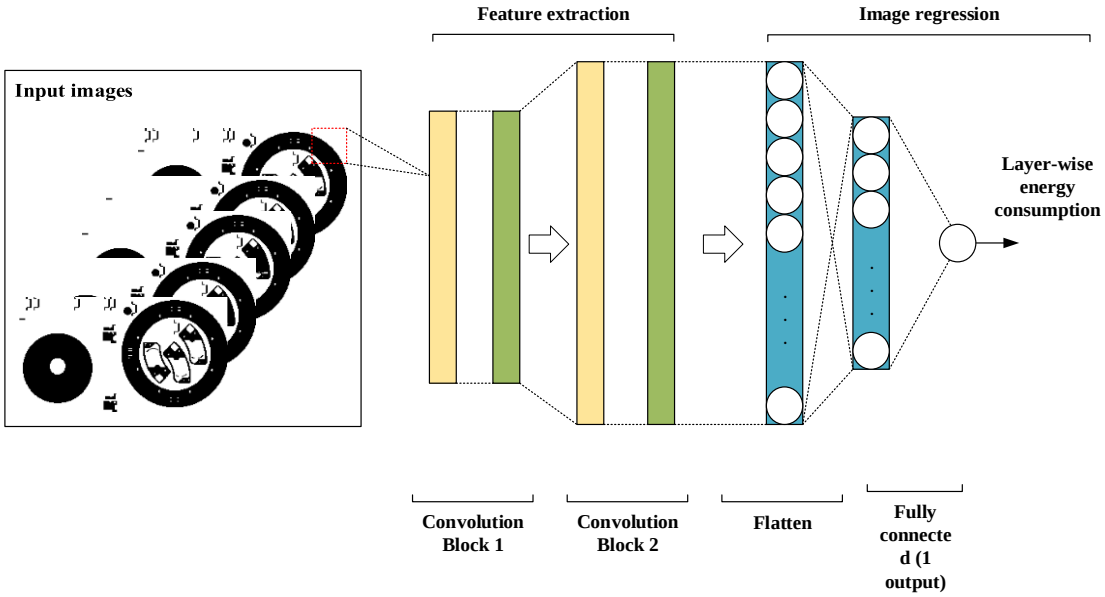


Figure 5.2 Student model architecture overview.

- ***CNN Architecture on the Targeted FPGA Platform***

Figure 5.3 shows the redesign of the student network for the targeted FPGA platform. The parameters of the distilled student network are further quantised to fit the FPGA platform. Input image data and CNN parameters are stored in multiple Block Random Access Memories (BRAMs) instead of a traditional buffer module. Using Vivado IP cores, these BRAMs integrate into the FPGA design, facilitating access to the

parameters such as weights and biases crucial for convolutional operations. It is unnecessary to add intermediate buffers thereby leading to a reduction of latency and energy consumption. The parallel processing capabilities of the FPGA are leveraged in convolution operations. The design efficiently manages convolution and pooling layers through loop unrolling, which optimises both operations by adapting to predefined sizes.

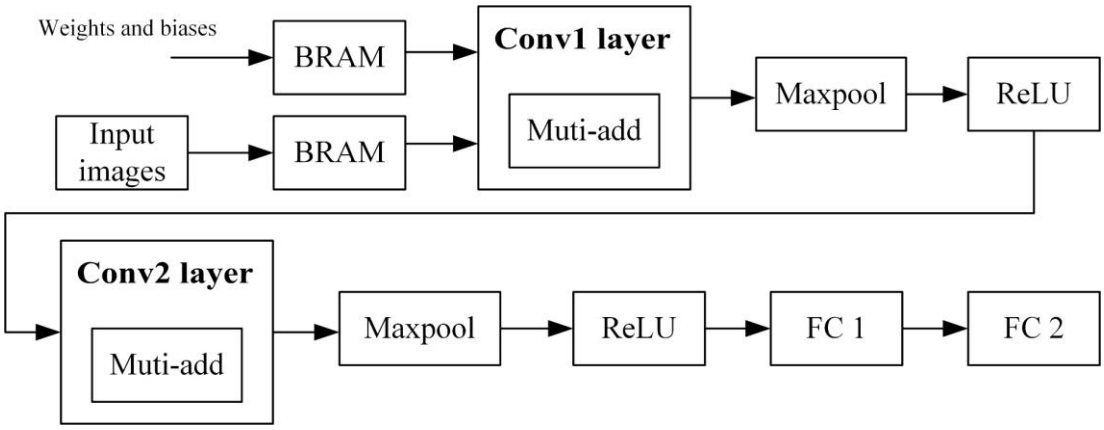


Figure 5.3 Block diagram of a CNN designed for FPGA deployment and acceleration.

A Finite State Machine (FSM) simplifies the workflow of control and computation phases of a CNN by controlling the sequence of operations. In energy-sensitive environments where consistent performance is critical, this approach minimises complexity and ensures reliability. Convolutional layers integrate seamlessly with max pool layers, followed by ReLU activation, and fully connected layers. By integrating these components, the minimised latency can be realised, significantly reducing power consumption. The design eliminates the need for intermediate buffers and directly utilises BRAM for instant access to the weights and biases of the CNN, thereby reducing the convolution process to an efficient Multiply-Accumulate (MAC) operation. The convolutional layer processes the input directly to the max pooling layer, followed by ReLU activation and finally through the fully connected layer.

During convolution, input data are multiplied by the weights of their corresponding filters, and then the results, along with biases, are summed. CNNs perform convolutional computations using MAC operations. Convolution is performed by sliding the convolution kernel (or filter) over the input data and multiplying it with the data at each position according to a specific pattern. The filter (or kernel) slides over the input data, multiplying its elements with the covered input data elements using a small weight matrix. A complete feature map is generated by accumulating each multiplication result into a single output value across the entire input. Figure 5.4 demonstrates the layout of MAC operation in the convolutional layer when being accomplished in FPGA.

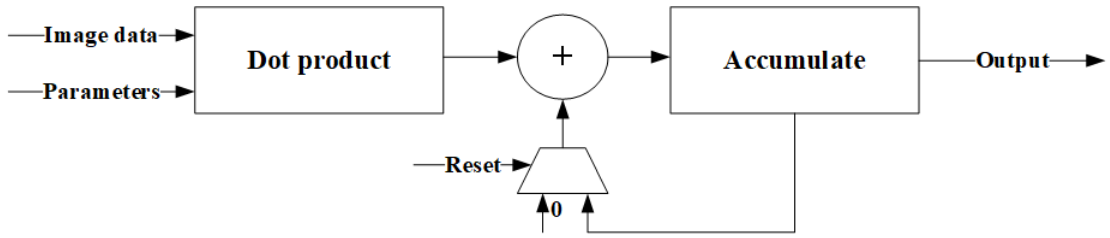


Figure 5.4 Fundamentals of convolutional operation within CNNs.

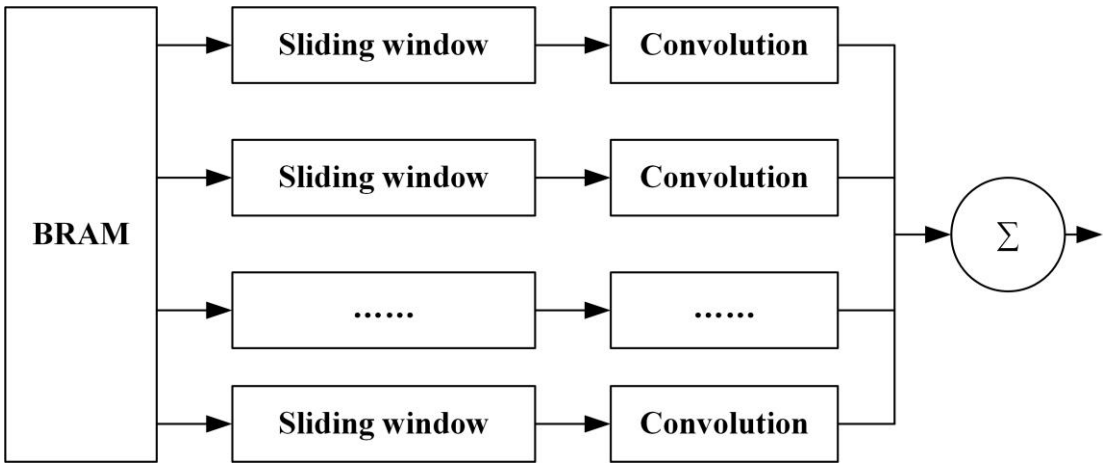


Figure 5.5 Structural architecture of a convolutional block in CNNs.

The sliding window technique allows convolutional kernels to be created with a very low memory bandwidth requirement. Figure 5.5 shows the convolution process, highlighting how each pixel is stored in FPGA memory for multiple uses. In this scenario, each filter involves n multiplications followed by $n-1$ additions. As the windows sliding applies over the image, output feature maps are generated by repeatedly calculating the dot product of 5×5 input data with 5×5 filter parameters (weights and biases). This process involves storing input data in a buffer that processes data within the convolutional layer. Multiple sliding windows, each representing a convolution operation, are moved over the buffer in parallel to extract local features. Convoluted feature maps are the output of each sliding window.

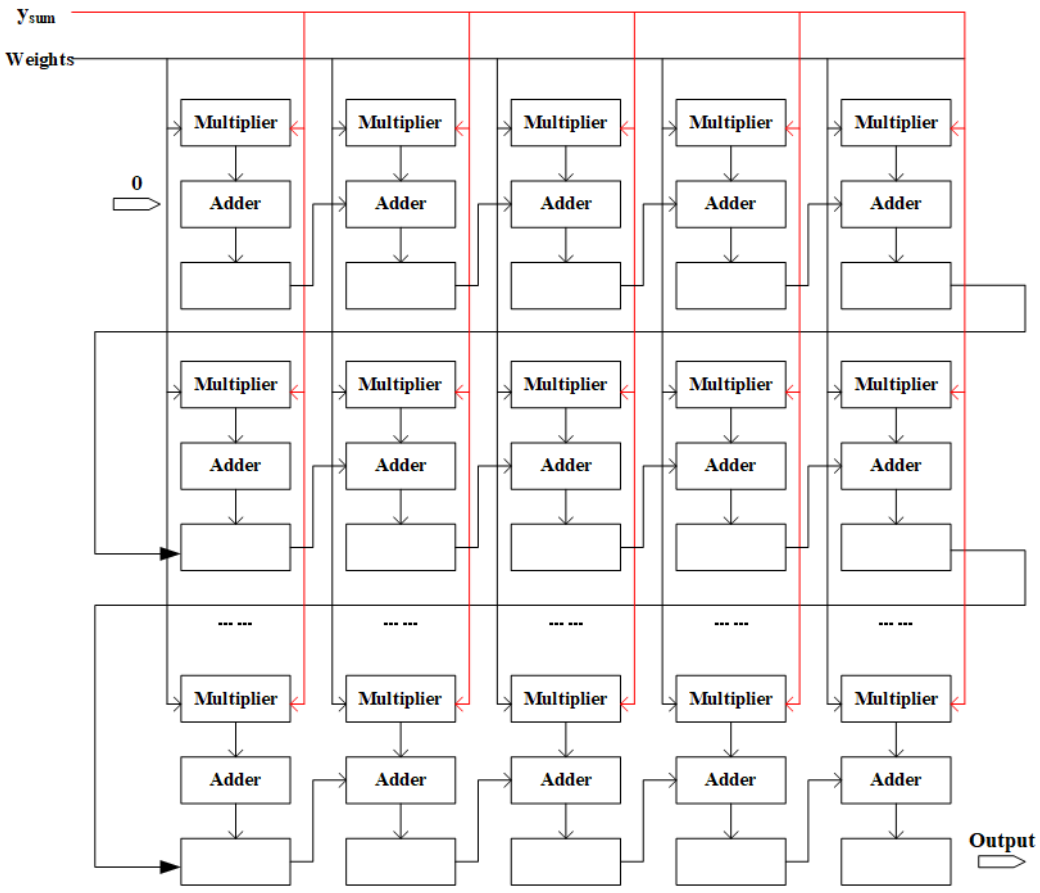


Figure 5.6 Detailed structure of a 5×5 convolution operation including multiplications and accumulations.

Figure 5.6 demonstrates that multiple sliding windows and convolution operations are parallelised, allowing simultaneous processing of data regions. The networks then combine information from all feature maps at this layer for use in subsequent layers by aggregating the outputs of all convolution operations.

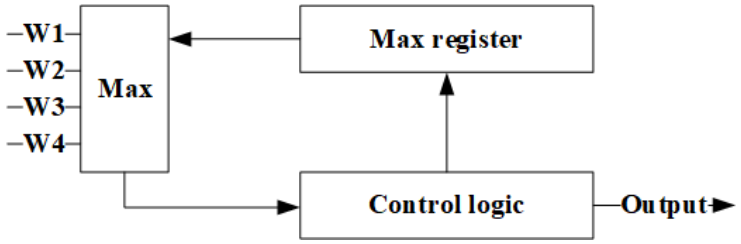


Figure 5.7 Maxpooling layer implementation in CNNs.

Comparatively, the max-pooling module selects the maximum value from the four input weights and determines the output based on this maximum value. If the value is zero, output the original value. In the ReLU module, the function is simply computed as $\boldsymbol{f(x) = \max(0, x)}$. Figure 5.7 illustrates the operations of the maxpooling module.

5.3 Experimental Design and Setups

5.3.1 Experiment Setups

- ***Deep Learning-based Energy Consumption Prediction Model***

This study started by employing an FPN and SPP architecture to model unit energy consumption on the layer level, leveraging image-based features. A significant advantage of the proposed architecture was its capability for multi-scale feature fusion. In the initial stage, layer-wise images were input into the proposed architecture. The

multi-scale feature fusion model served as the teacher model, which guided the training process of the student model. To evaluate the performance of the proposed model, it was compared against conventional image-based models.

- ***Lightweight Model through Feature-based Knowledge Distillation***

The role of the KD technique was to compress the complex model and transfer the generalisation capacity to obtain the smaller, simpler model. The KD shows its merits in reducing complexity and computational requirements to enhance applicability for acceleration on the targeted FPGA platform. The teacher model in the proposed methodology referred to a combined architecture including FPN and SPP modules for multi-scale feature fusion, while the student model network was a CNN architecture. In this experiment, the logit-based KD strategy was compared to identify the effectiveness of the feature-based KD strategy. After the KD process, the student was quantised to meet the resource requirements of the targeted FPGA platform.

- ***FPGA-accelerated Lightweight Model Development***

After KD and quantisation, the student model was accelerated by the targeted FPGA platform. The Xilinx ZYNQ-Z2 development board as shown in Figure 5.8 is used in this experiment to implement 13300 programmable logic elements and 220 Digital Signal Processing (DSP) units. The simulation was conducted using CNN functional modules, including buffers to store input pixels and calculations to process convolution operations, which were then inputted into Vivado for synthesis based on resource allocation, time constraints, and routing considerations.

Figure 5.8 PYNQ-Z2 as the targeted platform.

5.3.2 Evaluation Metrics

Table 5.1 Evaluation metrics.

Evaluation metrics	Equations
Root mean squared error (RMSE)	$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(p_t - a_t)^2}$
Mean absolute error (MAE)	$MAE = \frac{\sum_{t=1}^{N}\lvert p_t - a_t\rvert}{N}$
Model correlation coefficient (MCC)	$MCC = \frac{S_{PA}}{\sqrt{S_P S_A}}$ $S_{PA} = \frac{\sum_i (p_i - \bar{p})(a - \bar{a})}{n-1}$; $S_P = \frac{\sum_i (p_i - \bar{p})^2}{n-1}$; $S_A = \frac{\sum_i (a_i - \bar{a})^2}{n-1}$

Table 5.1 shows the equations for calculating RMSE and MAE. Notice that $\boldsymbol{a_t}$ is the actual value, and $\boldsymbol{p_t}$ is the predicted value based on the model. In addition to measuring the magnitude of error, both evaluation metrics can be used to diagnose variations in error in a set of predictions. Equations are provided for calculating RMSE, MAE and

MCC. It has been observed that the actual value differs from the predicted value resulting from the model. Since both evaluation metrics measure the magnitude of error in predictions, they can be used to diagnose error variations. Besides, MCC measures the correlation between actual values and predicted values on a range of -1 to 1 from negative to positive impact. As for FPGA, several performance metrics incorporate throughput, latency, resource utilisation and power consumption.

5.4 Results and Discussions

5.4.1 Performance of Teacher Model

This study first explored the potential for compressing DL algorithms, particularly CNNs, to predict energy consumption in AM. This study analyses layer-wise image data, focusing on design-relevant information to inform the prediction process.

Table 5.2 Comparative analysis of benchmarks and proposed CNN on AM data training performance.

	RMSE (Wh/g)	**MAE (Wh/g)**	**MCC**
MobileNet-V1	3.95	2.17	0.73
MobileNet-V2	3.22	1.33	0.92
EfficientNet	3.98	3.02	0.61
ShuffleNet	4.46	3.35	0.67
SqueezeNet	9.73	8.92	0.77
Vanilla CNN (6 layers)	3.71	2.9	0.81
Proposed Architecture (FPN+ SPP)	**2.87**	**1.74**	**0.78**

Table 5.2 compares the various lightweight network architectures, evaluating model correlation with MCC and precision with RMSE and MAE. The proposed multi-scale feature fusion architecture leverages EfficientNet as the backbone and shows its merits in efficiency and accuracy. It has the lowest RMSE (2.87 Wh/g) and the second-highest MAE (1.74 Wh/g). Additionally, it has a moderate MCC of 0.78, which indicates that it is very robust in capturing potential trends in the data. The findings reveal the importance of the proposed methodology, identifying the advanced ability of an effective balance in terms of accuracy and efficiency. The integration of the FPN and SPP module in the proposed architecture effectively captures the hidden multi-scale features from the layer-wise images, benefiting from feature fusion and predictive modelling.

Table 5.3 Comparative analysis of benchmarks and proposed CNN based on FLOPs and parameters.

	FLOPs	**Params**
MobileNet-V1	11.82M	3.21M
MobileNet-V2	103.97M	2.21M
EfficientNet	8.27M	739.5K
ShuffleNet	22.83M	76.96K
SqueezeNet	77.18M	722.69K
Vanilla CNN (6 layers)	101.49M	4.27M
Proposed CNN	**15.23M**	**272.53K**
KD student	**1.71M**	**19.62K**

As a major component of the methodology, a relatively large, comprehensive multi-scale feature fusion model is trained and then a KD process to transfer its knowledge to a smaller, more deployable student model. This process involves reducing larger network parameters while preserving valuable information. Consequently, the smaller model maintains a similar level of performance despite having fewer parameters. The advantage of this model is that it is lighter, more flexible, and easier to deploy. Based

on the results, it can be concluded that the simplified student model has reduced complexity and fewer parameters but maintains a similar level of accuracy as the teacher model. Table 5.3 compares the FLOPs and the number of parameters of the student network after the KD strategy and the existing lightweight network. The results show that the student model with feature-based KD has the lowest number of parameters and a reduced number of FLOPs. In contrast to the direct design of a new architecture, the feature-based KD involves training a smaller network based on the knowledge from the teacher network, which achieves robust performance with a significantly reduced number of parameters. This makes it an ideal option for deployment on edge platforms such as FPGAs, where efficiency and low resource consumption are crucial.

5.4.2 Results of KD Strategies

The findings in Table 5.4 indicate that feature-based KD shows its effectiveness in capturing complex patterns through advanced feature utilisation. Although there is a similar pattern in the logit-based KD strategy with a lower mean error, potentially leading to overfitting due to the increasing variability. Despite its simplicity and low computational requirements, a common 2-layer CNN that is not distilled has poor accuracy and consistency.

Table 5.4 Effectiveness comparison of knowledge distillation variants in terms of plain, logit and feature-based strategies.

	RMSE (Wh/g)	MAE (Wh/g)	MCC
Plain (2-layer CNN)	3.85	3.39	0.74
Logit-based KD	4.41	3.26	0.74
Feature-based KD	**3.63**	**2.81**	**0.75**

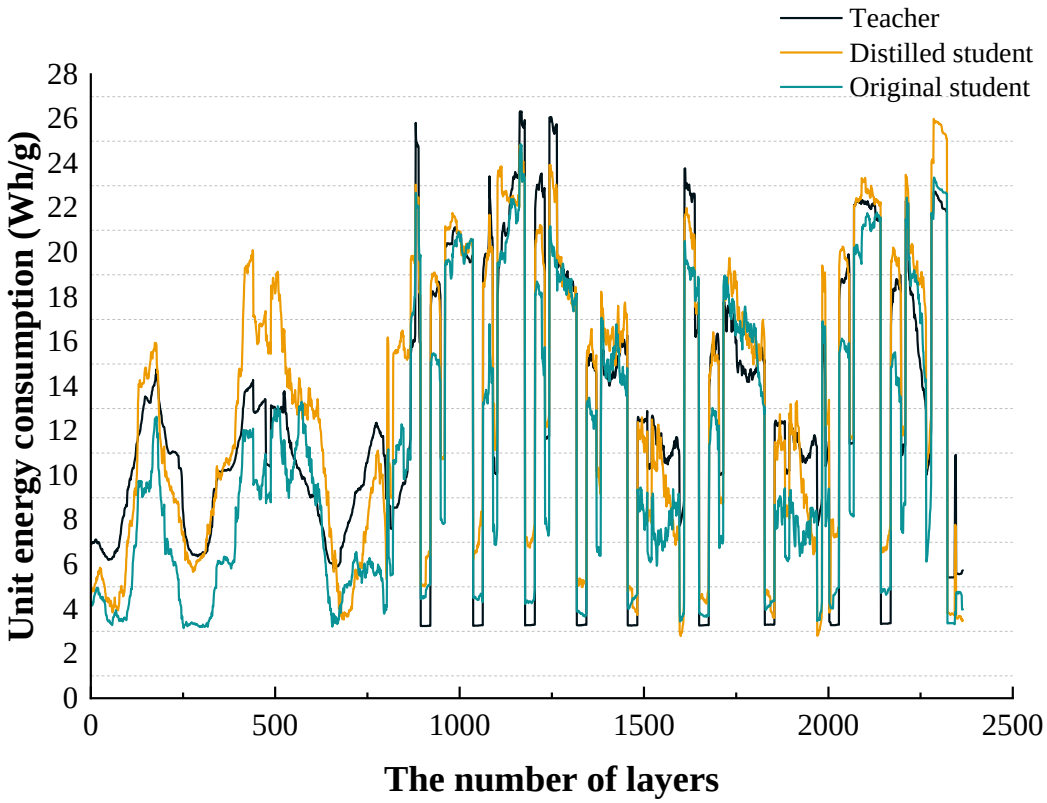


Figure 5.9 Comparative predictive performance of teacher, distilled student and original student models.

Figure 5.9 illustrates a comparison of the prediction performance of the proposed multi-scale feature fusion architecture (teacher model), the student after-distillation process, and the original two-layer CNN that is not distilled. It can be observed that the changes in energy consumption values are significant because the original student model has less generalisation ability to the new and unpredicted data. While employing the KD, the trend of value tends to be smoother and nearly matches that of the actual value. According to the previous results, the KD process has shown its merits.

5.4.3 Performance of Implementation of FPGA-CNN

Table 5.5 and Table 5.6 show the resource allocation and power consumption. Analysis reveals that logic and computation-intensive tasks in FPGA deployment require a

significant number of LUTs and DSP modules. However, FFs and I/O modules are underutilised at 6% and 21%, respectively, suggesting the need for additional functionality or interface complexity for optimisation. This study indicates that dynamic usage accounts for the majority of power consumption at 68%, with signal and DSP modules contributing to the highest percentage at 35% for both. This result suggests a need for improved power management strategies. There is potential for improving FPGA designs, especially in the areas of power and resource management.

Table 5.5 Synthesis resource utilisation report.

	Available	Utilisation	Utilisation %
LUT	15664	53200	29
FF	6363	106400	6
LUTRAM	120	140	/
DSP	104	220	47
I/O	26	125	21

Table 5.6 Power utilisation metrics

	Resource	Power Consumption (W)
Dynamic (68%)	Clocks	0.017 (7%)
	Signal	0.081 (35%)
	Logic	0.051 (22%)
	DSP	0.080 (35%)
	I/O	0.001 (1%)
Static (32%)	PL static	0.108 (100%)

With limited resources in the targeted FPGA deployment and acceleration, the design must be optimised to meet these constraints. KD can produce a simplified CNN version that maintains the original model's performance while meeting FPGA resource

constraints. These calculations can be significantly improved by utilising an FPGA platform. The parallelism of FPGA shows its merits in processing large-scale convolution operations. The integrated structure is used, integrating the different layers and functions of a CNN rather than treating them as separate modules. The results indicate that the data transfer between different layers of the CNN can be performed efficiently inside the FPGA, reducing the latency and energy consumption associated with data transfer. Besides, the integrated structure simplifies the CNN architecture, reducing the complexity of the interface between different modules. With low latency and parallel processing capabilities, FPGAs are ideally suited to handle large volumes of data effectively and analyse in real-time. In AM processes, various sensors monitor the operation of the machine, including the consumption of energy. FPGA devices process this data rapidly and analyse it using CNN models. Based on patterns and trends influencing the consumption of energy, these models can identify energy-intensive layers in manufacturing processes.

Part designers and process operators are expected to consider the energy consumption associated with specific parameter settings regarding part design and process planning. These parameters play an important role in effective AM design. The FPGA platform contributes to offering design and production strategies, enabling the design process more energy efficient. Different from traditional processing methods, the FPGA platform can potentially reduce the latency of image processing and feature fusion in the future, which is critical for handling large amounts of image data. The integration of FPGA and DL techniques contributes to a significant impact on AM systems. In addition to accurate predictions and energy consumption monitoring, this robust integration enables AM energy consumption to be predicted efficiently and effectively, as well as offers insights into decision-making through data analysis.

5.5 Summary

To sum up, this chapter focused on the effects of the integration of DL and the Xilinx PYNQ-Z2 FPGA platform to predict energy consumption in an SLS system. The

proposed methodology combines a multi-scale feature fusion technique and feature-based KD to train a lightweight model. After the distillation process, the lightweight model significantly reduces its parameters while maintaining satisfactory performance. This lightweight model shows robust performance with reduced computational requirements. This compression technique facilitates the efficient processing of layer-wise image data, and the feature-based KD strategy has been successfully implemented for FPGA deployment. The allocation of FPGA resources is reflected in the balanced utilisation of LUT, DSP, and BRAM by the convolutional, pooling, and fully connected layers. The findings highlight the potential benefits of integrating CNN structures on FPGAs, characterised by lower resource utilisation and reduced power consumption. This highlights the advantages of FPGA-based implementations.

In the following chapter, the proposed approach will move onto an FPGA-based monitoring and management system to optimise the energy consumption prediction model based on design-relevant data, image-based data and energy-relevant data within an SLS system.

Chapter 6 FPGA-based Management System for Optimising Energy Consumption in SLS

6.1 Introduction

There has been worldwide recognition of the problems associated with establishing energy-relevant models in AM systems (Baumers et al. 2011). These energy-relevant models in the detection of excessive energy utilisation at the specific layers, facilitate the adjustment of process parameters for planning, design and operations before or in the process. This enhances the quality of the product and the overall process (Tian et al. 2019a). FPGA platforms are often reprogrammable to perform new types of computing tasks. This is due to the computing capabilities and sufficient flexibility, which also allow them to work collaboratively with CPUs in terms of training and inference, thereby accelerating the computing tasks in the dynamic manufacturing environment (Singh and Gill 2023).

There is a need for establishing an energy consumption optimisation system based on the collaboration of CPU and FPGA for this specific task driven by the desire to improve efficiency and reduce the carbon footprint of SLS systems (Kellens et al. 2014). Developing the energy consumption prediction models is an important component in optimising AM systems, and plays a key role in analysing and managing the AM data for decision-making and support for manufacturers (Watson and Taminger 2018) before exploring potential improvement options for different designs. DfAM focuses on the functionality and manufacturability of the final parts followed by energy efficiency as part of the design considerations (ALMASRI et al. 2022). Based on the principle of DfAM, it is possible to integrate data-driven systems to optimise the AM system, for example, by considering energy consumption before the design phase (Chinchanikar and Shaikh 2022). The precise information required to optimise designs is provided by energy consumption modelling (Qin et al. 2022). DfAM helps manufacturers consider energy efficiency at the design stage, resulting in more

economical and environmentally friendly manufacturing through the energy consumption management of several design options (Vaneker et al. 2020).

The dynamic environment of AM processes makes it difficult to predict and manage energy consumption, so conventional approaches to modelling energy utilisation may not be sufficient for the dynamic environment of AM systems (Saimon et al. 2024). Methods that rely on predefined models are often inadequate to manage the complex data produced by AM systems (Liu et al. 2018b). As advanced structures and features are integrated into designs, the complexity of the design increases. Hence, there is a need for advanced technologies that can adapt to the changing environment of SLS processes and provide real-time optimisation. DL offers a promising alternative, capable of learning autonomously from data, continuously enhancing its predictions as it is exposed to diverse layer-wise images and associated energy consumption (Alzubaidi et al. 2021). However, existing DL models are parameter-intensive (Chen et al. 2021), leading to the challenges of direct deployments on the edge device (Cheng et al. 2017). A further improvement on lightweight models is considered.

Building on the methodology and findings in Chapter 5, this chapter further explores a comprehensive approach to FPGA-based management and optimisation support, including multi-scale feature fusion, lightweight model design, and the PSO technique. The proposed approach is expected to support the decision-making process of the design, in which a data-driven management and monitoring system can optimise parameter settings for minimal energy usage by leveraging PSO and DL, allowing for continuous improvement in the SLS process. Such integration can pave the way for the development of more sustainable and cost-effective SLS systems. The proposed approach will contribute to more sustainable and cost-effective manufacturing practices. The remainder of this chapter presents the technical overview of the DL-based optimisation system for SLS. The following section details the experimental setup based on a real-world SLS scenario. After that, the last section validates the proposed approach and reports the results, accompanied by a discussion.

6.2 Method of Enhancing Predictions with FPGA-CNN

Figure 6.1 illustrates the overview of an FPGA-based management system for optimising energy consumption in an SLS system. In the beginning, the data was collected from an AM machine including design-relevant data, energy-relevant data and layer-wise images. After the data processing stage, the U-net architecture with CBAM and AC serves as the teacher model. CBAM improves feature representation, but the DNN architecture that it employs for energy consumption prediction models often demands significant computational and memory resources, requiring the compression and acceleration of these models without sacrificing performance.

This method utilises a dual KD strategy, leveraging logits and intermediate features to train a smaller student network within a simplified architecture for energy consumption prediction, transferring knowledge from a larger teacher network to a smaller one and preserving a lightweight structure. Additionally, parameter quantisation reduces model complexity and computational demand, making it ideal for inference and training acceleration on the FPGA. This, combined with other model compression techniques, creates compact, high-performance models essential for resource-constrained devices and environments. The student mode's feature fusion can be accelerated by the targeted FPGA. These features and predicted energy consumption values will be integrated with another DNN for the PSO to optimise the design-relevant parameters to minimise the energy consumption of the selected builds.

The student model, integrated with the PSO algorithm on the targeted FPGA platform, predicts build energy consumption based on part-design and process-planning parameters. Optimised parameters and unit energy consumption of the build are expected to provide decision support and inform designers and operators before the process. Currently, the method integrating CPU and FPGA platforms can collaboratively collect image data from CAD models. Leveraging features from these image data, the predictive model extracts valuable insights from the historical data,

which is then trained collaboratively with the PSO algorithm. This allows designers to collect and analyse design-relevant data before the additively manufactured parts start.

The method aims to optimise the design process, facilitating a more cost-effective and sustainable design cycle. Future work will focus on an FPGA-CNN in the real-time environment to predict energy consumption, potentially leading to more energy savings and increasing operational efficiency. The predictive models provide valuable insights from both real-time and historical data, aiding in the optimisation of quality control and predictive maintenance in AM processes. Manufacturers can identify and adjust AM machine process plans by monitoring energy consumption patterns in real-time.

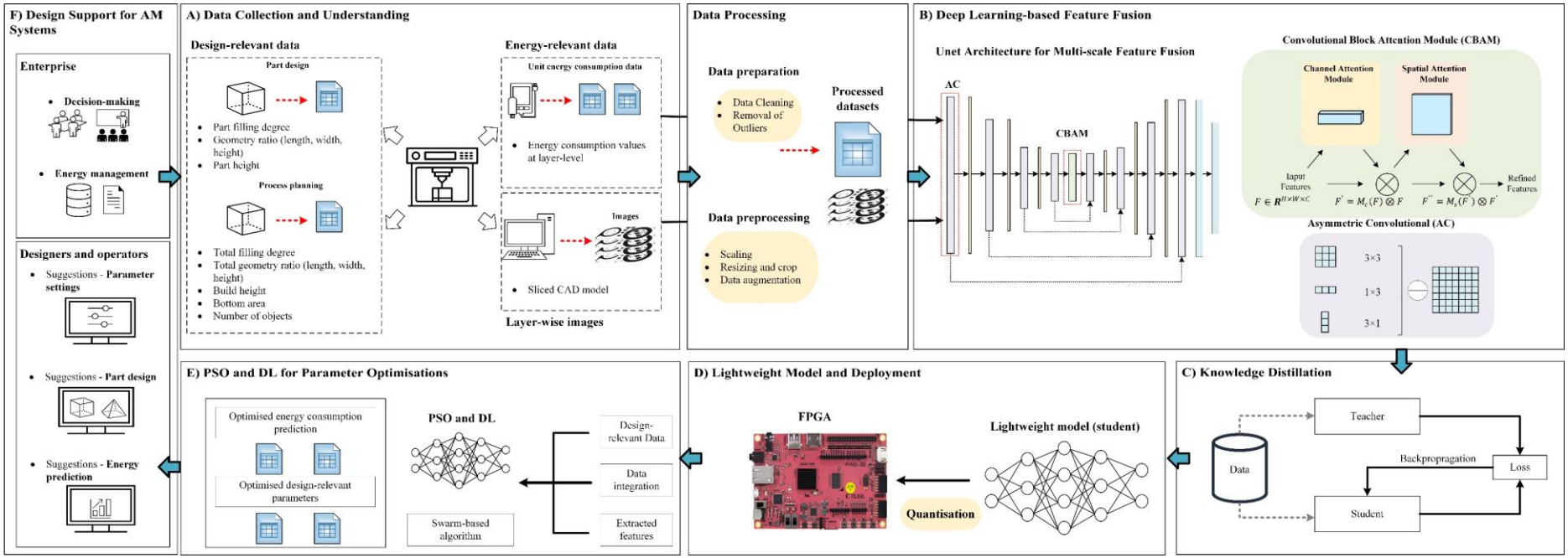


Figure 6.1 Overview of energy consumption modelling in AM monitoring with design parameters, image features, and energy-relevant data.

6.2.1 Multi-scale Feature Fusion for Energy Consumption Predictive Modelling

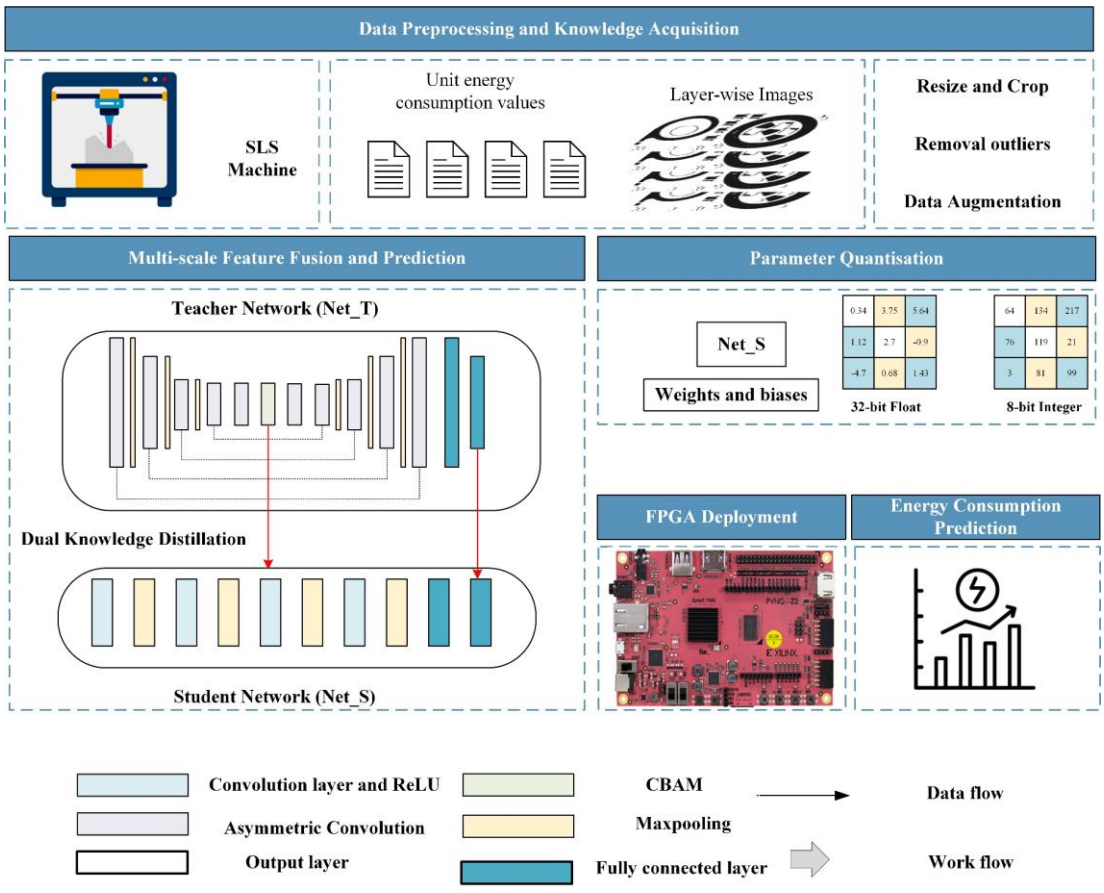


Figure 6.2 Detailed workflow of FPGA-based predictive modelling for SLS energy consumption.

This first stage was related to developing the teacher model for multi-scale feature fusion on layer-wise images. Building on the development in Chapter 5, this approach extended the work to another more efficient architecture. A U-Net was considered as the teacher, to which the modification of the architecture was applied. Compared to the conventional architecture, the proposed architecture incorporated convolutional layers replaced by AC layers and the CBAM in the bottleneck. In an SLS scenario, part geometrical information will have features related to direction, such as rectangular shapes or patterns in specific directions. AC can capture these features or insights.

CBAM, as one of the attention mechanisms, can focus on local features of the layer-wise images such as edges, holes, dense filling areas, etc. CBAM can leverage the attention mechanism to concentrate more on those local features, providing more accurate predictions. By channel and spatial attention, the model can present the ability to process both local and global information in feature maps, which makes the model handle complex geometry and design in a more robust manner. In the broader context, the proposed model serves as the teacher model which is also a lightweight model by reducing computational complexity and burden. This may contribute to training student models before deployment.

The dual KD strategy utilised each feature representation from each layer in the encoder to obtain the multi-scale features and the logit from the output of the teacher network. Figure 6.2 depicts the energy consumption predictive modelling on the FPGA platform. This workflow belongs to the part of the entire approach that contributes to enhancing predictive modelling by using FPGA-based predictive modelling. The following sections will illustrate the detailed works, beginning with multi-scale feature fusion for energy consumption modelling, followed by the employment of the dual KD technique, lightweight energy consumption model and FPGA collaboration. The proposed architecture employs the AC block and CBAM. The details are described in the following sections:

- ***Enhancing U-Net Performance through Asymmetric Convolution Block***

The AC block splits traditional filters into horizontal and vertical components, each with two layers that measure m×1 and 1 × m, to reduce the network's computing complexity and parameter counts (Tian et al. 2022). Two-dimensional kernels have a rank of 1 and can be equivalently converted into one-dimensional convolutions (Ding et al. 2019). Equation (6.1) describes the output channel feature map at the j-th filter, where $\boldsymbol{M}$ is the input channel feature map, $\boldsymbol{F}$ is the k-th input channel of $\boldsymbol{F}^{(j)}$. AC decomposes the convolution kernel into two separate convolution operations ($\boldsymbol{H} \times \boldsymbol{W}$) in both vertical and horizontal directions, respectively.

$$O_{:,:,j} = \sum_{k=1}^{C} M_{:,:,k} * F_{:,:,k}^{(j)} \tag{6.1}$$

In this experiment, the teacher network combines 3×3, 1×3, and 3×1 convolution kernels to extract image features. The 3×3 kernel captures local and global features, while the 1×3 and 3×1 kernels focus on the details in the horizontal and vertical directions, respectively. In addition to the robustness of this multi-scale feature extraction approach, the AC block reduces parameters and computation, thus improving the accuracy and efficiency of the energy consumption prediction model. In contrast to the conventional U-Net architectures, which typically employ conventional convolution layers, the proposed architecture employs AC layers. This hybrid approach improves efficient processing and better feature representation, facilitating energy consumption prediction through layer-wise images.

- ***Convolution Block Attention Module***

The CBAM includes channel and spatial attention (Woo et al. 2018), which exploits features from two dimensions to obtain attention maps for adaptive feature refinement. The following equations illustrate the overall attention process, which $\boldsymbol{M_c(F)}$ and $\boldsymbol{M_s(F')}$ represent the attention maps for channel and spatial attention respectively.

$$F' = M_c(F) \otimes F \tag{6.2}$$

$$F'' = M_s(F') \otimes F' \tag{6.3}$$

Adding a weighting mechanism within the feature map after the convolutional layer enables CBAM to demonstrate the significance of the feature extraction. It improves the representation of essential features, enabling the model to capture details in the sliced images better. Integrating CBAM into the U-Net architecture can significantly improve feature representation by refining the feature map through an adaptive

attention mechanism. Furthermore, this attention mechanism, CBAM, can effectively filter out irrelevant noise in the images, which makes it helpful to predict layer-wise energy consumption and improve the effectiveness and robustness of the predictive model.

- ***The Proposed Architecture of U-Net for Energy Consumption Predictive Modelling***

U-Net provides a more compact architecture, with skip connection preventing information loss and maintaining important features. It effectively integrates multi-scale features due to its encoder-decoder architecture, as used in biomedical segmentation (Ronneberger et al. 2015). The skip connections in this architecture play a vital role in merging high-resolution features from earlier layers with the low-resolution features from deeper layers to enhance capturing details. The image data in SLS needs to consider local details and global geometry, which U-Net leverages skip connections to preserve high-resolution features. Obtaining insights from the traditional architecture and functions of U-Net, the proposed U-Net architecture aims to provide efficient image segmentation and multi-scale feature fusion, effective for 128×128 input dimensions in layer-wise images.

The architecture starts with four encoding blocks. Differently from conventional architecture, each convolutional layer is replaced by an AC layer, and the number of channels in these layers increases progressively through each encoding block from 1 to 32, 32 to 64, 64 to 128, and 128 to 256. After the encoding block, there is a bottleneck layer that connects the encoder to the decoder, employing CBAM to focus on important features by utilising channel-wise and spatial-wise attention mechanisms. The bottleneck remains 256 channels which compresses the information into a more manageable size to preserve most critical features. Subsequently, the decoding path up-samples the feature maps and it restores the spatial dimensions of the feature maps. After each up-sampling step, the concatenation of feature maps allows the network to combine high-resolution features. The final output layer includes a 1×1 convolution,

followed by global average pooling to reduce the spatial dimensions of the feature maps to a single value per channel. A fully connected layer is employed last.

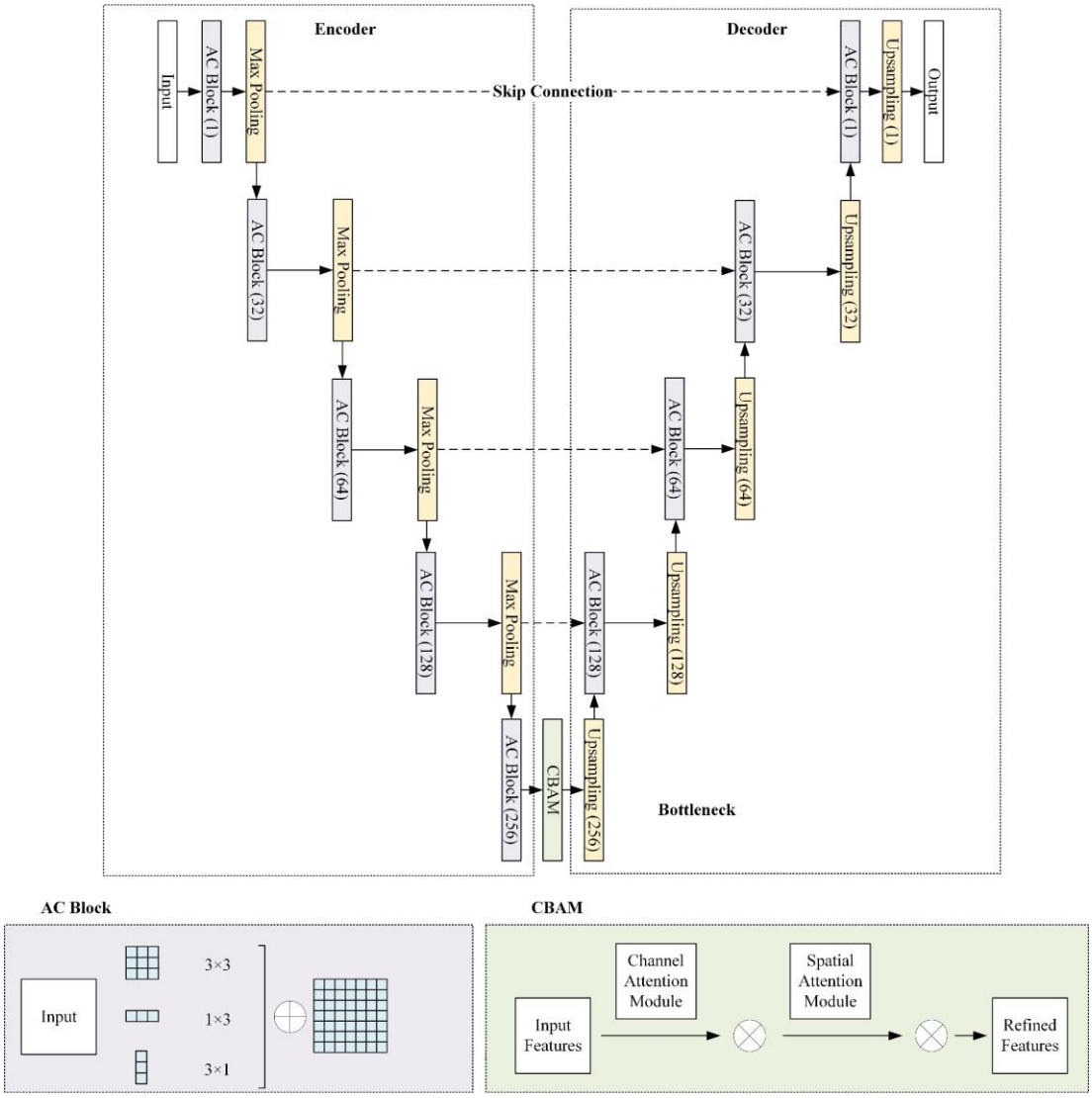


Figure 6.3 Teacher model architecture for multi-scale feature fusion.

6.2.2 Dual KD Strategy

The dual KD strategy combines feature and logit-based knowledge. It provides rich feature representations of the encoder and the final output. Combining these two strategies improves the learning efficiency and performance of the student network.

Equation (6.4) provides the loss function of the proposed dual KD strategy, combined with distillation loss $\boldsymbol{L_{distil}}$ from logit, actual label loss $\boldsymbol{L_{label}}$ and loss from the feature layer $\boldsymbol{L_{feature}}$. By experiment, the optimal hyperparameter $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ controls the weight of each loss.

$$L_{total} = \alpha \cdot L_{distil} + (1 - \alpha) \cdot L_{label} + \beta \cdot L_{feature} \tag{6.4}$$

In Equation (6.5), $\boldsymbol{L_{distill}}$ represents the difference between the student and teacher network's logits (predicted outputs) using the L1 loss.

$$L_{distill} = Smooth_{L1}(f_s(x), f_t(x)) \tag{6.5}$$

where Equation (6.6) defines L1 loss

$$Smooth_{L1}(x, y) = \phi(x) = \begin{cases} \frac{1}{2}(x - y)^2 & \text{if } |x| < 1 \\ |x - y| - \frac{1}{2} & \text{otherwise.} \end{cases} \tag{6.6}$$

According to Equation (6.7), the distillation loss of the feature layer determines the mean squared error between the intermediate feature maps of the student model and the teacher model, thus providing the student model with the feature representation as the teacher model, where the feature $\boldsymbol{f_{s,i}(x)}$ and $\boldsymbol{f_{t,i}(x)}$ are the $\boldsymbol{i}$-th feature maps for the student and teacher.

$$L_{feature} = \frac{1}{N}\sum_{i=1}^{N}\left(f_{s,i}(x) - f_{t,i}(x)\right)^2 \tag{6.7}$$

In Equation (6.8), the actual label loss $\boldsymbol{L_{label}}$ measures the difference between student predictions and the actual labels by using L1 loss.

$$L_{label} = Smooth_{L1}(f_s(x), \mathrm{y}) \tag{6.8}$$

It is important to note that these equations define a comprehensive loss function for training the student model using the dual KD technique. This technique enables the student to learn from the teacher model's outputs and its internal representations of features.

6.2.3 Lightweight Energy Consumption Model and FPGA Collaboration

The role of convolution blocks is critical to extract features, which selects filter sizes incrementally. Teacher models typically consist of multiple convolutional layers, pooling layers, as well as a fully connected layer. Figure 6.4 illustrates the structure of the student model, which comprises four convolutional layers, one ReLU activation layer, four pooling layers, and two fully connected layers. The features of input images are extracted by the convolutional, activation and pooling layers, while the fully connected layers implement the regression analysis. In the context of regression analysis, the output layer does not require applying a softmax activation function. In contrast, the regression model only requires the output layer to generate a continuous value corresponding to the predicted output.

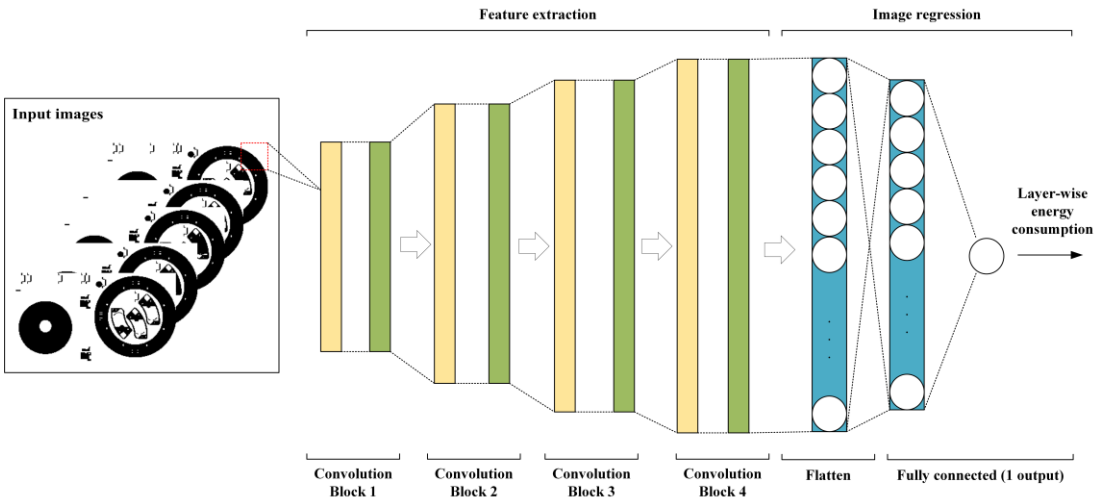


Figure 6.4 Student model architecture overview.

Parameters of the student network are quantised to deploy on the targeted platform. Due to the significant reduction in computation and complexity in the student model, the student model may not generalise as well as the teacher model. The KD process is thus used to mitigate the differences in model complexity between the teacher and student models. For embedding in a small computing platform, a student model has a smaller scale and is less complex than teacher models. A student model is more efficient on smaller computing platforms due to its less memory cost and computation while maintaining sufficient structure to capture the features of images. By using four convolution blocks and two fully connected layers with one output layer, the student model focuses on the most important features of the data. As a result, lower-precision weights and biases are produced on convolutional and fully connected layers in comparison to the teacher model.

Before integrating the student model with the targeted FPGA platform, the parameter counts require further quantised to reduce the computational complexity and memory requirement due to the resources available on the targeted FPGA platform. The quantisation technique is essential in further compressing model complexity, as it approximates the representation of a DL model using floating point numbers by a model using low-bit width numbers. As a result of quantisation, fewer bits are required

to represent weights and biases in a CNN, preventing the shortage of storage and memory, as well as the computation complexity. Therefore, the student model can be accelerated in terms of feature extraction by the targeted FPGA. These features and predicted energy consumption values will be fused with a deep neural network for the PSO to optimise the design-relevant parameters to minimise the energy consumption of the selected builds.

6.3 PSO-based Technique for AM Parameters and Energy Optimisation

After employing the FPGA and energy consumption model collaboratively, the output features and predicted energy consumption of the unique layers can be integrated into another DNN for data integration. Additionally, this DNN leverages PSO to optimise the best combination of design-relevant parameters including part design and process planning to find the minimal energy consumption of the build. These findings are anticipated to support and guide the decision-making for part designers and process operators before the manufacturing process in an SLS system. This section introduces the hybrid PSO-based technique for AM parameter optimisation and energy consumption prediction.

6.3.1 Research Visions and Solutions in FPGA and Predictive Model Integration

Figure 6.5 shows the energy consumption management system of the FPGA-CNN in AM. With FPGA-based CNNs, predictive analytics and decision-making are transformed into AM energy consumption management. In the offline training phase, the predictive analysis is conducted before the fabrication. Deep CNN models can be implemented to offer predictions to avoid the potential increase in energy consumption. These models are trained from the historical layer-by-layer CAD model before the build begins. At the real-time monitoring phase, the pre-trained and distilled model is deployed FPGA. This setup enables the implementation of AM systems performing

real-time layer-wise image preprocessing, which facilitates the extraction of hidden and significant features by CNNs in a more rapid manner.

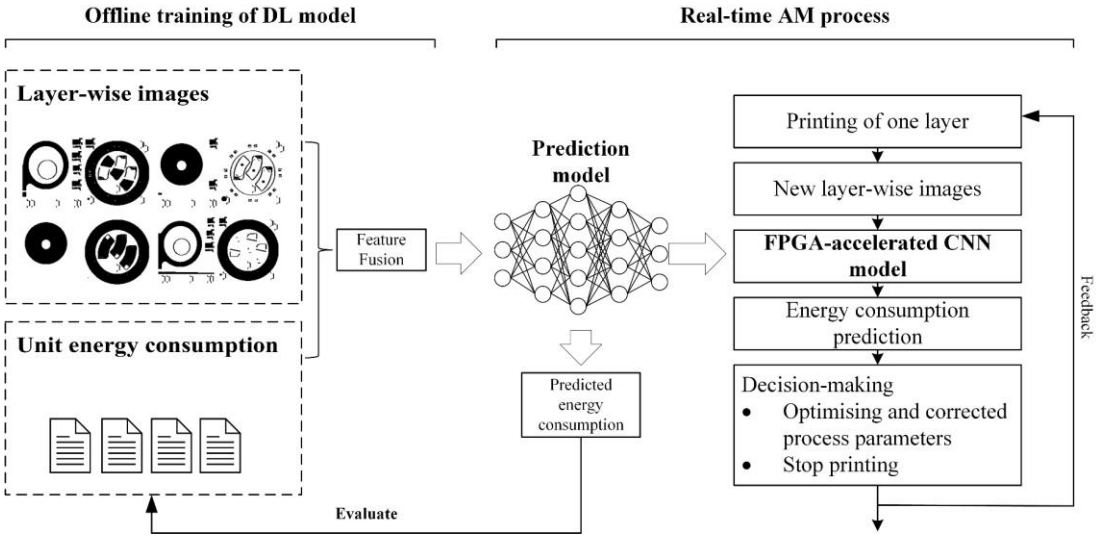


Figure 6.5 Integrated solution of FPGA-based CNNs for AM energy consumption analysis.

The FPGA-CNN system monitors the energy-relevant data against the predictions, allowing for optimising parameters before the build starts if high energy consumption is anticipated. An FPGA-CNN can be used to take timely action to predict energy consumption immediately leading to significant energy savings and increased operational efficiency. The predictive models provide valuable insights from both real-time and historical data, helping optimise quality control and predictive maintenance in AM processes. Manufacturers can identify and correct the inefficiency by monitoring energy consumption patterns in real-time, which may involve adjusting parameters or pausing the process. However, in the current scenario, the integration of FPGA and DL into AM can assist designers in the design process and offer operators more cost-effective operations in the manufacturing process based on the optimised parameter settings regarding part design and process planning.

6.3.2 PSO Technique in the SLS System

Figure 6.6 describes the optimisation process by using PSO algorithms. The PSO algorithm employs a population of particles to simulate the search for an optimal solution. Each particle represents a potential solution whose position and velocity are continuously adjusted within the search space. Particles adjust their movements according to their historical individual optimal position and the global optimal position found by the swarm, thereby guiding the search process towards minimising energy consumption.

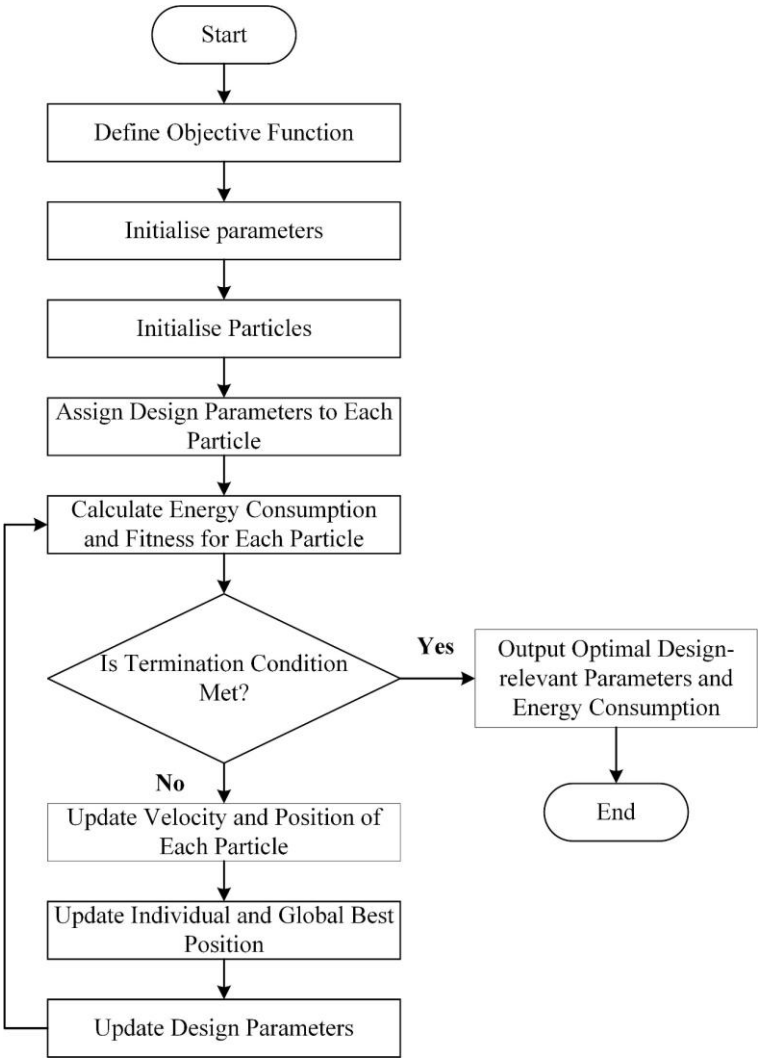


Figure 6.6 Integration of PSO in the energy consumption prediction model.

Firstly, the objective function $\boldsymbol{f(X)}$ is defined as follows: $\boldsymbol{E}$ is the predicted energy consumption. $\boldsymbol{S}$ represents the image-based feature dataset of the selected samples. $\boldsymbol{D}$ represents the image-based feature dataset of the selected samples. This is shown in Equation (6.9).

$$f(X) = E\big(S, D(X)\big) \tag{6.9}$$

$$X = [x_1, x_2, \dots, x_n\,] \tag{6.10}$$

In this equation, $\boldsymbol{x_1, x_2, \dots, x_n}$ correspond to the different design-relevant parameters such as part filling degree, part rate on width and length dimensions etc. The objective function $\boldsymbol{f(X)}$ serves to evaluate the performance of different combinations of design parameters by calculating the total specific energy consumption of the model for a given set of parameters. When the design parameters are distributed to each particle, the PSO algorithm parameters are configured. $\boldsymbol{v_{id}}$ is the is the velocity of the particle in a D-dimensional space without volume and mass. $\boldsymbol{w}$ represents an inertia weight to control the inertia of particle velocity. $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ are the acceleration coefficients, while $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are the random numbers ranging from 0 to 1 (Yao et al. 2024). Equation (6.11) illustrates the mechanism for updating the particles' velocities.

$$v_{id} = w \cdot v_{id} + c_1 \cdot r_1 \cdot \big(P_{best_{id}} - x_{id}\big) + c_2 \cdot r_2 \cdot (G_{best} - x_{id}) \tag{6.11}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{6.12}$$

By updating iteratively, the algorithm efficiently explores the solution space before identifying the optimal combination of design-relevant parameters that are expected to reduce energy consumption. After the initial setup and configuration of the PSO algorithm, particles are randomly initialised with positions and velocities within the search space. Utilising this fitness function $\boldsymbol{f(X)}$, particles iteratively adjust their positions and velocities, progressively converging towards their best individual position ($\boldsymbol{P_{best}}$) and global position ($\boldsymbol{G_{best}}$), thereby optimising their search strategy to achieve the goal of finding optimal solutions. According to Equations (6.11) and (6.12), the velocity and position of each particle are updated for the next iteration. Equation (6.12) indicates the update of the position of the new particles.

When the process reaches the maximum number of iterations or the global optima, the process ends. When the global best position ($\boldsymbol{G_{best}}$) is determined, the optimal solution and the corresponding design and process planning parameters can be obtained. The PSO algorithm efficiently navigates through the complex search space, which iteratively enhances the solution until it terminates.

6.4 Experimental Design and Setups

Combining design-relevant parameters with image data to optimise overall energy consumption is a potential approach to achieve efficient design. Using PSO with the proposed multi-scale feature fusion and KD strategies identifies the optimal set of parameters to minimise energy consumption values. Design parameters such as filling degree, part ratios, part height and bottom area play a crucial role in determining the energy consumption of a product. These parameters define the physical and operational characteristics of the design, directly affecting energy utilisation. In this case study, design parameters are optimised based on their impact on energy consumption, which is predicted by a trained DL model.

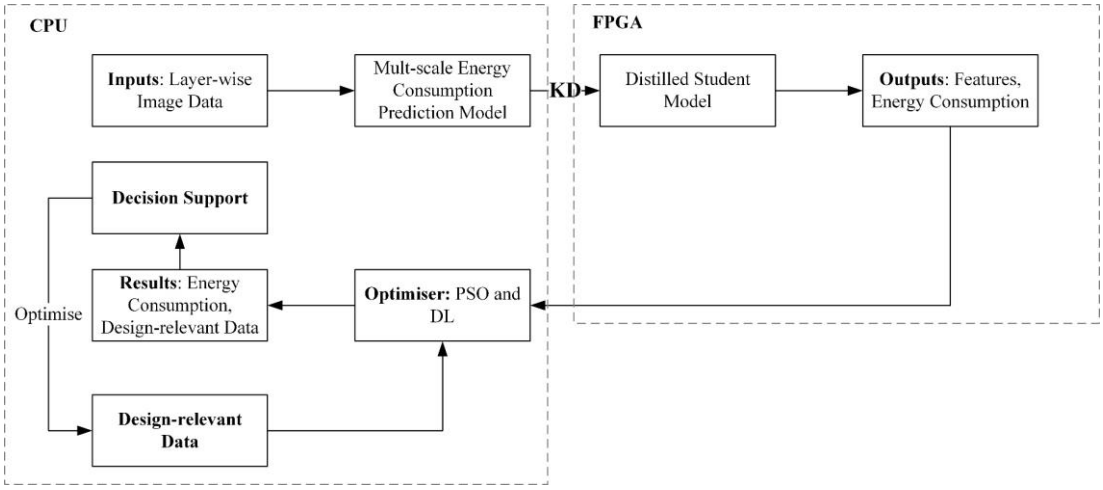


Figure 6.7 Experimental setup for SLS energy optimisation monitoring.

Figure 6.7 illustrates the experimental setup for energy optimisation in the case study. The overall method comprises two main sections: the CPU and the FPGA. On the CPU side, the process began with the pre-processing of layer-wise image data. Subsequently, the process data was input into a trained energy consumption prediction model, which employed a KD strategy to obtain the lightweight version for deployment on the targeted FPGA platform. The FPGA accelerated the execution, providing predicted energy consumption values for each layer. After that, features of the multi-scale energy consumption prediction model could be extracted based on the layer-wise images. Image data is a critical input to the energy prediction model, enabling it to capture the detailed geometric features of the design. Each layer-wise image preserves essential information such as geometry and orientation. Processing this image data through multiple layers of convolution and pooling, the model extracts and fuses high-level features that are indicative of the design's energy efficiency.

Based on these predicted values and the features, the lightweight energy consumption model worked with an optimiser, utilising the PSO algorithm to optimise a set of design parameters that minimised the specific energy consumption of a process, as predicted by the student network. The parameters included geometric and operational characteristics of the system, such as filling degree, part ratios, bottom areas and total

height. The PSO algorithm explores the multi-dimensional design space by iteratively adjusting these parameters and evaluating their impact on energy consumption. The output of the PSO process is an optimised combination of design parameters that minimise energy consumption. These optimisation parameters are of great value to designers in helping them understand the most efficient shape proportions and other key design choices. For instance, an optimal aspect ratio or filling degree may indicate a design that balances structural integrity while minimising energy consumption. Providing designers with these optimisation parameters, the methodology can support informed decision-making during the design process. Part designers and process operators can use these optimised parameters to create more energy-efficient designs and processes. The method integrates prediction and optimisation capabilities to improve the energy efficiency of SLS processes, demonstrating a data-driven approach to industrial energy management in SLS systems.

6.4.1 Experiment Setups

- ***DL-based Energy Consumption Prediction Model***

The first experiment was associated with teacher network training. Two data types correspond to layers, including image and specific energy consumption. Removing the outliers and resizing was required to clean the dataset and standardise the image dimensions. Due to the number of small layer-wise images, data augmentation was vital in increasing the model's generalisation ability while avoiding overfitting. After that, the data was input into the proposed feature extraction and fusion architecture. The backbone of the proposed approach was U-net architecture, which had enhanced feature extraction ability and the CBAM to focus on the most relevant regions of the input image data, thereby improving feature extraction. The AC blocks were integrated with traditional U-net architecture to capture spatial features more efficiently, which helped improve the receptive field and reduce computational complexity. The experiment aimed to achieve superior feature extraction and data fusion using an enhanced U-net architecture that combined an attention mechanism and asymmetric

convolutional blocks. This approach aimed to provide more accurate predictions and better generalisation from training data to new unseen datasets.

A comparative experiment was conducted between the traditional U-net and enhanced models, training on the AM dataset and evaluating the performance of the RMSE, MAE and MCC. Further details regarding these evaluation metrics can be found in Section 6.4. This experiment aimed to demonstrate the merits of enhanced U-net architecture offering more accuracy and better generalisation from training data to new data than traditional U-net and CNN.

- ***Lightweight Model by KD***

KD is a key technique for compressing the model into a smaller and simpler architecture, thereby reducing complexity and computational requirements, and making it suitable for acceleration on the FPGA platform. During the KD process, the teacher network was a U-net that employed multi-scale feature fusion, and the student network was a CNN designed to learn and predict unit energy consumption at the layer level. Experimentation with various KD strategies, including feature and logit-based approaches, leads to the adoption of a student model that integrates both feature and logit-based strategies. Following the KD process, the parameters of the student model were further quantised through the quantisation technique to ensure it was compatible with the targeted FPGA.

The experiment aimed to develop a lightweight model using the KD strategy, leveraging both logit and feature-based distillation techniques in a dual KD approach. This approach combined the advantages of both techniques to improve the performance of the student model. This student model architecture comprised four convolutional layers and two FC layers for feature extraction and output of the predicted values. The dual KD approach potentially provided a balanced transfer of final output and features from each encoder in U-net. The comparative analysis was

conducted based on the comparison between SOTA lightweight architectures and distilled student architecture, as well as the ablation study regarding AC blocks and different KD strategies.

- ***Lightweight Model Acceleration with the Targeted FPGA Platform***

After KD and quantisation, the student model was accelerated by the targeted FPGA platform. The Xilinx PYNQ-Z2 FPGA platform as shown in Figure 6.8 is used in this case study with 13300 programmable logic elements and 220 Digital Signal Processing (DSP) units.

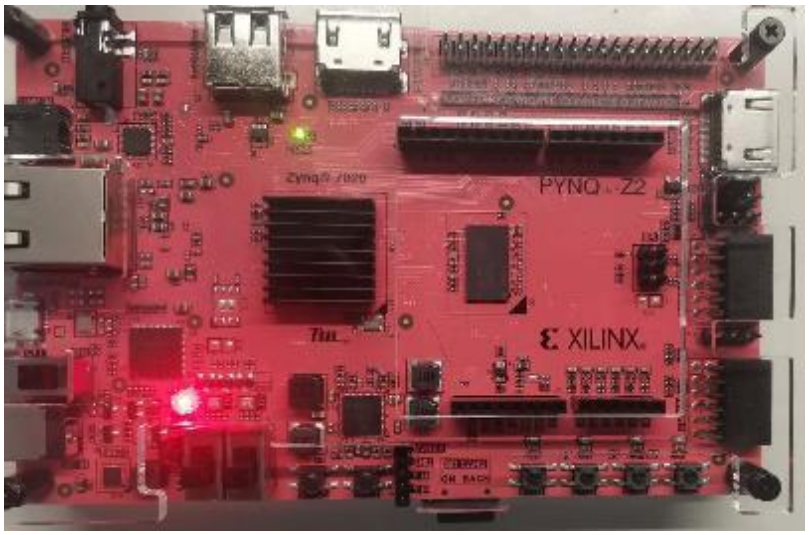


Figure 6.8 Xilinx PYNQ-Z2 platform for data processing in the experiment.

Figure 6.9 shows that, due to the heterogeneous nature of the Zynq7020, the design utilises Programmable Logic (PL) for various image-based tasks, such as acquisition, caching, processing, and output. The regression function of the student network was implemented on the Processing System (PS) which communicated the recognition results back to the PL side via the AXI-Lite bus, facilitating interaction between the software and hardware components of the system. Architecture development was managed through Vivado 2023.2 for simulation, synthesis and implementation, and

the programming language was Verilog. To manage the interaction between the FPGA and the laptop, the bitstream file and overlay were downloaded from Vivado and uploaded to the targeted FPGA platform. The targeted FPGA platform supported coding on Jupyter Notebook, enabling direct code development and testing on the PYNQ-Z2 platform. The experiment aimed for high performance and efficiency, evaluating by resource utilisation, throughput, latency and power.

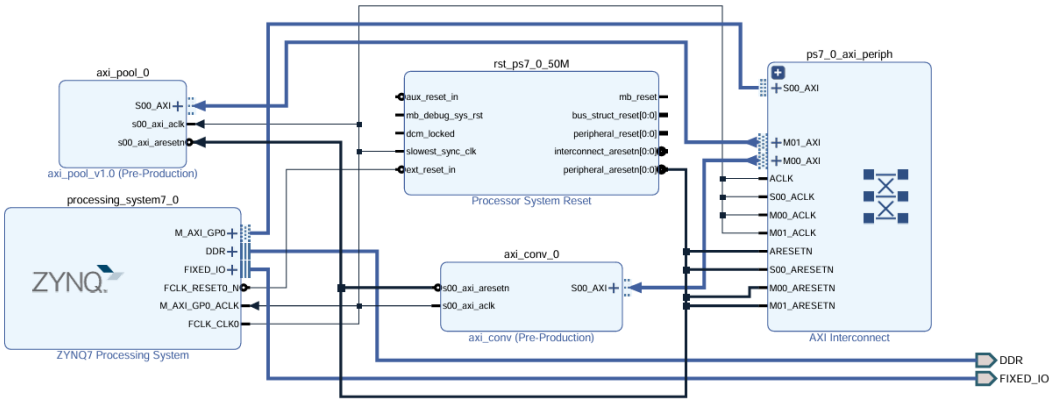


Figure 6.9 The overlay of the project for the experiment.

- ***PSO-based optimisation technique on design-relevant parameters***

Based on these predicted values and the features, the lightweight energy consumption model worked with an optimiser, utilising PSO to optimise a set of design parameters that minimise the specific energy consumption of a process, as predicted by the student network. The parameters consisted of design-relevant data on the building level, such as filling degree, part rotation and position, bottom areas and total height. PSO plays a critical role in providing the optimal parameters iteratively to minimise energy consumption, thereby evaluating their impact on energy consumption. When the algorithm reached its termination condition, it provided the optimal parameters corresponding to the lowest energy consumption value. These optimal parameters were valuable to part designers and process operators for determining the most effective

geometry and other key design selections, such as the optimal part ratio between length and width or part filling degree before the process.

In the experiment, the optimised and original parameters were compared, indicating the modifications of the parameters. These changes lead to a reduction in energy consumption, which could therefore support informed decision-making in the design process by providing designers with these optimised parameters. Designers could use the insights to create more energy-efficient designs, whether in other SLS machines, where energy consumption is a key concern. This system integrated prediction and optimisation to enhance energy efficiency in SLS processes, demonstrating a data-driven approach to industrial energy management in SLS systems.

6.4.2 Evaluation Metrics

Table 6.1 The evaluation metrics.

Evaluation metrics	Equations
Root mean squared error (RMSE)	$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(p_t - a_t)^2}$
Mean absolute error (MAE)	$MAE = \frac{\sum_{t=1}^{N}\lvert p_t - a_t\rvert}{N}$
Model correlation coefficient (MCC)	$MCC = \frac{S_{PA}}{\sqrt{S_P S_A}}$ $S_{PA} = \frac{\sum_i (p_i - \bar{p})(a - \bar{a})}{n-1}$; $S_P = \frac{\sum_i (p_i - \bar{p})^2}{n-1}$; $S_A = \frac{\sum_i (a_i - \bar{a})^2}{n-1}$

Table 6.1 demonstrates the evaluation metrics for determining the performance of the proposed architecture. The performance can be evaluated using three key metrics: RMSE, MAE and MCC, which provide comprehensive insights into the accuracy and reliability of the model.

As for FPGA, several performance metrics incorporate throughput, latency, resource utilisation and power consumption. Evaluating the performance of an FPGA deployment involves several key metrics that help identify the efficiency, effectiveness, and practicality of the deployment. These metrics cover various aspects including speed, resource utilisation and power consumption. Latency, defined as time, is critical for this implementation requiring high-speed data processing. Throughput, measured in operations per second, indicates the amount of data processed by the FPGA per unit time, with higher throughput reflecting better performance for data-intensive applications. Resource utilisation refers to the percentage of FPGA resources (such as logic blocks, DSP slices, BRAM, and I/O pins) used by the deployed design, and efficient utilisation ensures the FPGA can handle additional functionalities while reducing costs. Lastly, power consumption, measured in watts (W), is the amount of power consumed by the FPGA during operation. Lower power consumption is essential for battery-powered or energy-efficient applications and affects thermal management and cooling requirements.

6.5 Results and Discussions

6.5.1 Results of the Multi-scale Feature Fusion Model

Table 6.2 lists the proposed CNN architecture and baseline U-net and CNN results. It is observed that the vanilla U-net has the best performance in terms of RMSE at 2.93 Wh/g and MAE at 2.18 Wh/g due to its good feature extraction and data fusion. U-nets are often utilised in medical segmentation using the encoder-decoder architecture.

Unlike conventional CNN, it focuses on accurate feature extraction at the pixel level, which leaves more spatial information, increasing the performance. Due to the simple architecture of 6-layer CNN, they did not reach the expected performance. Besides, conventional CNN does not apply end-to-end training, leading to a high computational workload with 4.3M parameters involved, thereby decreasing the training efficiency. By implementing the AC block, MAE reduces to 2.03 Wh/g while RMSE increases to 3.31 Wh/g. Such fluctuations indicate that the AC block introduces changes to the model that improve its general accuracy while maintaining some prediction errors. The AC block assists in reducing computational workload, representing a decrease in the number of parameters by approximately 20%. The CBAM was implemented in the U-net architecture to improve the performance further. The RMSE and MAE at 3.06 Wh/g and 2.33 Wh/g have shown a slight increase in performance while still not reaching that in the conventional U-net. However, it reduces the parameters significantly compared to conventional U-net by about 75%. Reducing the number of parameters leads to lower computational and memory requirements, making them more suitable for training and deployment in resource-constrained environments, in which the model can train faster and perform inference more swiftly.

Table 6.2 The comparison of baseline and proposed CNN architecture.

Architectures	RMSE (Wh/g)	MAE (Wh/g)	MCC	FLOPs	#Params
Vanilla CNN (6 layers)	3.71	2.9	0.81	101.5M	4.3M
Vanilla U-net	2.93	2.18	0.87	386.9M	948.8K
U-Net with Asymmetric Convolution	3.31	2.03	0.71	285.6M	758.9K
Proposed Methodology (Attention + AC block)	**3.06**	**2.33**	**0.91**	**93.4M**	**236.2K**

6.5.2 Results of the Lightweight Model

Table 6.3 The ablation study of the proposed architecture and vanilla U-net with the KD process regarding different KD strategies.

Asymmetric Convolution	KD Strategy	RMSE (Wh/g)	MAE (Wh/g)	MCC	FLOPs	#Params
No	KD student (logit-based)	4.06	3.10	0.87	44.96M	148.55K
	KD student (feature-based)	4.99	3.20	0.91	93.36M	346.15K
	KD student (logit + feature)	**3.82**	**2.69**	**0.897**	**46.14M**	**181.58K**
Yes	KD student (logit-based)	3.01	2.50	0.86	44.96M	148.55K
	KD student (feature-based)	3.78	3.01	0.82	93.36M	346.15K
	KD student (logit + feature)	**3.77**	**2.65**	**0.892**	**93.36M**	**346.15K**

The findings in Table 6.3 compare the role of different KD strategies and the employment of AC block on both the proposed architecture and the vanilla U-Net. The findings demonstrate that the AC block enhances performance across different KD strategies, resulting in higher model performance. The dual KD approach without AC blocks leverages logit knowledge and intermediate feature representations. It outperforms the individual logit or feature-based approaches, achieving the lowest RMSE and MAE values of 3.82 Wh/g and 2.69 Wh/g, respectively. In addition, introducing AC blocks further improves performance, with an RMSE of 3.77 Wh/g and an MAE of 2.65 Wh/g when applying a dual KD strategy on the student model. The AC block boots feature extraction efficiency by employing asymmetric convolutions. It reduces computational overhead and model complexity, reflecting the reduced number of parameters. That means the distilled student model is more suitable for deployment in resource-constrained environments. These findings have significant

practical implications. Reducing RMSE and MAE leads to more accurate and reliable predictions, which is crucial for energy consumption prediction in AM systems. Moreover, the decreased computational load and model complexity enable faster training and inference, enhancing the model's scalability and usability in real-world scenarios.

Table 6.4 The comparison of SOTA lightweight architecture and distilled student network

Architectures	RMSE (Wh/g)	MAE (Wh/g)	MCC	FLOPs	#Params
MobileNet-V2	3.22	1.33	0.82	103.9M	2.2M
EfficientNet	3.98	3.02	0.61	8.3M	739.5K
ShuffleNet	4.46	3.35	0.67	22.8M	76.9K
SqueezeNet	9.73	8.92	0.77	77.2M	722.7K
Vanilla U-net	2.93	2.18	0.87	386.9M	948.8K
Student network	7.20	4.68	0.79	44.9M	148.6K
KD student (logit + feature)	**3.77**	**2.65**	**0.89**	**93.4M**	**346.2K**

Table 6.4 presents a comparison of the performance of different SOTA lightweight architectures and the distilled student network from the proposed methodology. While the KD student network (based on logit + features) does not outperform all the SOTA architectures, with an RMSE at 3.82 Wh/g and an MAE at 2.69 Wh/g), it achieves a competitive balance between performance and efficiency. This architecture achieves a good balance of low error rates and high computational efficiency, with 103.9M FLOPs and 2.2M parameters. After applying AC blocks, the KD student network (logit + feature-based) achieves a balanced performance with an RMSE of 3.77 Wh/g and an MAE of 2.65 Wh/g, representing a significant improvement over the student without AC blocks, as well as the baseline student network, with an RMSE of 7.20 Wh/g and an MAE of 4.68 Wh/g. The KD process enhances the student network's accuracy by

effectively transferring knowledge from the teacher network, improving its generalisation capabilities. These findings have important practical implications, particularly for applications requiring efficient deployment on resource-constrained devices. With its reduced computational and memory requirements, the KD student network is well-suited for scenarios where balancing performance and efficiency is crucial, such as edge devices and embedded systems. With 93.4M FLOPs and 346.2K parameters, it achieves competitive error rates while maintaining a relatively low computational burden.

6.5.3 Deployment of the Lightweight Student Model

The distilled student model is quantised before deploying on the targeted FPGA platform. Figure 6.10 illustrates the comparison of the actual and predicted values of the energy consumption of different samples. It can be observed that the difference still exists because of the effect of quantisation and the architecture of the FPGA platform.

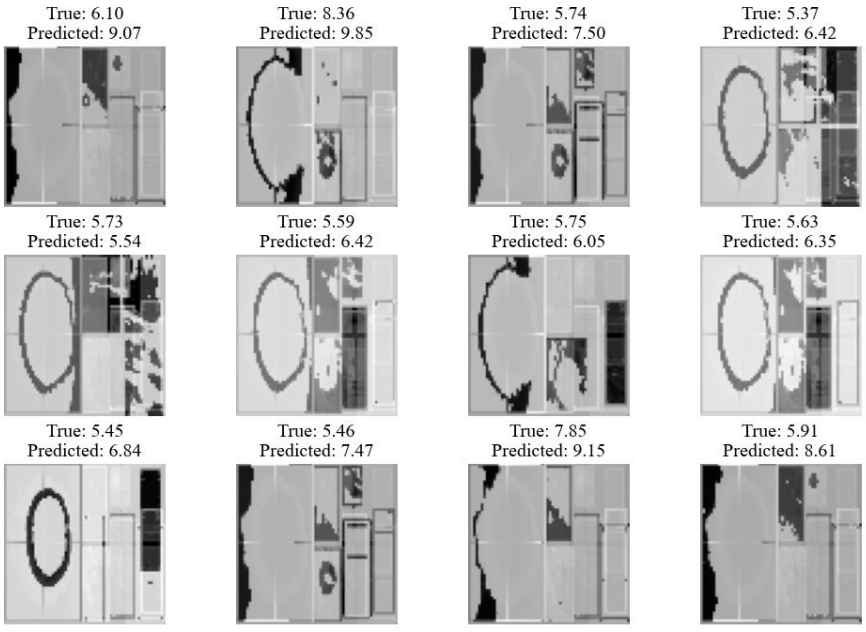


Figure 6.10 The actual and predicted energy consumption of samples.

As a summary of the FPGA resource allocation for the CNN in Table 6.5, key insights include the utilisation in convolutional layers (Conv_1 and Conv_2) as indicated by the increased LUT and DSP allocation, which are vital to extracting features from layer-wise images. Pooling layers (Pool_1 and Pool_2) demonstrate optimised design with lower LUTs and consistent DSP usage, which is crucial for maintaining data throughput. As a result of their role in the high-level integration of data, the fully connected layers (FC_1 and FC_2) show balanced resource use. A strategic allocation of BRAM, for the storage of results, emphasises the importance of efficient data handling, which is essential for the precision and efficiency required by AM systems. As a whole, this FPGA design exhibits a well-balanced balance between processing speed and accuracy, which is crucial for high-quality AM.

Table 6.5 Optimised CNN module utilisation report.

Module	LUT	BRAM Tile	DSP
Conv_1 (incl. ReLU)	3706	\	3
Conv_2 (incl. ReLU)	3944	\	4
Pool_1	582	\	4
Pool_2	867	\	4
FC_1	2451	\	3
FC_2	2429	\	2
BRAM (weights & biases)	24	8	\
BRAM (input)	11	4	\
BRAM (result)	43	16	\

Table 6.6 provides a detailed report on the synthesis resource utilisation of IP cores, highlighting that the FPGA implementation utilises 27% of LUTs, 3% of FFs, 20% of BRAM, 9% of DSPs, and 10% of BUFGs. These relatively low utilization percentages indicate significant potential for further optimisation and scalability. The low utilisation of FFs and LUTRAM suggests that the current design is efficient regarding flip-flops and memory resources. In contrast, the moderate utilisation of LUTs and DSPs points to areas where computational efficiency could be further enhanced.

Table 6.6 Detailed report on synthesis resource utilisation of IP cores.

Resource	Utilisation	Available	Utilisation (%)
LUT	14057	53200	27
FF	2818	17400	3
BRAM	28	106400	20
DSP	20	220	9
BUFG	12	32	10

The power utilisation metrics in Table 6.7 clearly distinguish between dynamic and static power consumption. Processing system (PS7) dominates the high dynamic power consumption at 1.528W, while other components such as clocks, signals, logic, and DSPs contribute minimally. The static power consumption is 0.136W, attributed to PL static power. These insights underline the importance of power efficiency in FPGA implementations, especially for applications requiring continuous operation on limited power sources. The high dynamic power consumption by PS7 suggests areas for potential power-saving optimisations.

Table 6.7 Power utilisation metrics.

	Resource	Power Consumption (W)
Dynamic (1.532W)	Clocks	0.002
	Signal	0.001
	Logic	0.001
	DSP	0.001
	Processing system (PS7)	1.528
Static (0.136W)	PL static	0.136

Table 6.8 compares the quantised student network's performance on the FPGA platform and CPU. The FPGA implementation shows higher RMSE (6.49 Wh/g), and MAE (5.80 Wh/g) compared to the CPU (RMSE of 3.06 Wh/g and MAE of 2.33 Wh/g). However, the FPGA significantly outperforms the CPU regarding power efficiency, consuming only 1.668W compared to the CPU at 38.7W. In addition, the FPGA achieves higher throughput (3051.8 frames/sec) and lower latency (327.7 ms) compared to the CPU (3210.2 frames/sec and 815.97 ms, respectively). The trade-off between accuracy and computational efficiency is critical in applications where power consumption and real-time processing are priorities. For example, FPGA implementation energy efficiency and high throughput in edge computing or embedded systems may outweigh the higher error rates.

Table 6.8 The comparison of FPGA and CPU in terms of running performance and power consumption (50MHz period).

	RMSE (Wh/g)	**MAE (Wh/g)**	**Throughput (frames/s)**	**Latency (ms)**	**Power (W)**
CPU	3.06	2.33	3210.2	815.97	38.7
FPGA	**6.49**	**5.80**	**3051.8**	**327.7**	**1.668**

6.5.4 Optimised Energy Consumption with Design-relevant Parameters and Image Features

Table 6.9 presents the original and adjusted build energy consumption for three different prototypes in the case study, by utilising different algorithms including Gradient Descent (GD), Genetic Algorithm (GA), Differential Evolution (DE), Simulated Annealing (SA), Bayesian Optimisation (BO) and PSO. PSO has shown its merits in reducing build energy consumption by adjusting the design-relevant parameters. The table compares adjusted values to original energy consumption to evaluate the effectiveness of each algorithm in reducing energy usage.

Table 6.9 Comparison of original and improved build energy consumption (Wh/g) for different algorithms.

Method	Build 1	Build 2	Build 3	Average Reduction
Original	376.66	365.60	287.69	/
Gradient Decent (GD)	391.82	380.86	297.88	-2.01%
Genetic Algorithm (GA)	314.03	386.76	257.59	10.6%
Differential Evolution (DE)	353.11	350.00	264.66	10.1%
Simulated Annealing (SA)	388.86	394.00	296.69	2.4%
Bayesian Optimization (BO)	397.83	398.83	298.43	-6.1%
Particle Swarm Algorithm (PSO)	**339.06**	**378.07**	**274.79**	**10.3%**

In the three build energy consumption and parameter optimisation, PSO for Build 1 is 339.06Wh/g, significantly lower than GD (391.82Wh/g) and SA (388.86Wh/g). For Build 2, the optimised energy consumption is 378.07Wh/g, which has achieved an increasing pattern compared to the original energy consumption. The optimised energy consumption in Build 3 is 274.79Wh/g, lower than GD (297.88Wh/g) and SA (296.69Wh/g) but higher than GA (257.59Wh/g) and DE (264.66Wh/g).

According to the average reduction of each algorithm, GA outperforms PSO in terms of average optimisation, with a 10.6% reduction, while PSO showed consistent optimisation across all builds, with a 10.3% reduction. Specifically, energy consumption in Build 2 increased significantly with GA and SA, while the pattern of optimised energy consumption by PSO demonstrates relatively fewer fluctuations. This indicates that PSO in this case study demonstrates a more stable optimisation performance. Overall, PSO achieves a relatively balanced performance and good optimisation effect across the three builds. The performance of BO in the three builds is also noticeable. The energy consumption of BO was 397.83 Wh/g in Build 1, 398.93 Wh/g in Build 2, and 298.43 Wh/g in Build 3. Compared to original energy

consumption values, the optimisation effect of BO in Build 1 and Build 3 is not significant, with an increase in Build 2, which denotes relatively poor stability when providing optimisation support for design-relevant parameters in the case study.

By combining the image-based features from the FPGA platform and design-relevant parameters by using PC, the PSO and DL model is applied to achieve the optimisation of design-relevant parameters, thereby minimising the energy consumption of the build. The following tables present the experiment data on optimised parameters including part design and process planning and the energy consumption values by using the proposed method. It provides a comparison across three different builds of design-relevant parameters on build-level and energy consumption after employing the PSO and DL-based approach. The main objective is to leverage the proposed approach to reduce the unit energy consumption (in Wh/g) and optimise those design-relevant parameters. What stands out in this table is the decrease in unit energy consumption in Build 1 and Build 3 from 376.66Wh/g to 339.06Wh/g and from 287.69Wh/g to 274.79Wh/g, respectively. There was no decrease in unit energy consumption based on the optimised parameters in Build 2, which means the combination of the parameters is less efficient for this build.

Table 6.10 Results of design-relevant parameters in Build 1 by using different optimisation algorithms.

Build 1	Original	GD	GA	DE	SA	BO	PSO
Degree of Part Filling (%)	12.03	12.47	10.58	23	13	10.05	**11.62 (↓0.41)**
Part Ratio (WL) (%)	1.06	1.11	1.06	1.36	1	1.163	**1.05 (↓0.01)**
Part Ratio (HL) (%)	0.61	0.61	0.72	0.55	0.70	0.69	**0.63 (↑0.02)**
Part Ratio (WH) (%)	1.73	1.46	1.65	3	1.5	1.55	**1.63 (↓0.01)**
Part Height (mm)	106.51	105.26	106.02	40	109.73	108.36	**107.78 (↑1.27)**
Degree of Total Filling (%)	11.17	10.66	11.38	4.76	10	12.14	**13.61 (↑2.44)**
Total Ratio (WL) (%)	0.55	0.48	0.54	0.10	0.5	0.48	**0.48 (↓0.07)**
Total Ratio (HL) (%)	0.54	0.60	0.55	0.2	0.6	0.514	**0.55 (↑0.01)**
Total Ratio (WH) (%)	1.02	1.97	1.13	2	1	1.01	**1.45 (↑0.43)**
Bottom Area (cm2)	2585.51	2410.37	2701.29	2000	2388.72	2675.05	**2628.42 (↑42.91)**
Height (mm)	371.02	401.72	386.62	103.58	408.06	354.54	**407.75 (↑36.73)**
Num of Part	24	26.82	39.57	12.01	31.41	32.38	**30.65 (↑6.65)**

Table 6.11 Results of design-relevant parameters in Build 2 by using different optimisation algorithms.

Build 2	Original	GD	GA	DE	SA	BO	PSO
Degree of Part Filling (%)	17.59	17.28	16.37	1.12	15.61	16.63	**16.74 (↓0.85)**
Part Ratio (WL) (%)	1.31	1.22	1.20	1.10	1.2	1.39	**1.32 (↑0.01)**
Part Ratio (HL) (%)	1.68	1.31	1.54	0.6	1.8	1.16	**1.24 (↓0.44)**
Part Ratio (WH) (%)	0.78	0.66	0.67	1.58	1	0.51	**0.69 (↓0.09)**
Part Height (mm)	188.5	179.04	177.44	105.67	183.84	179.04	**191.56 (↑3.06)**
Degree of Total Filling (%)	9.35	10.08	8.86	15	5.90	7.17	**9.04 (↓0.31)**
Total Ratio (WL) (%)	0.53	0.61	0.54	0.56	0.5	0.56	**0.6 (↑0.07)**
Total Ratio (HL) (%)	0.82	0.79	0.69	0.57	0.5	0.7	**0.7 (↓0.12)**
Total Ratio (WH) (%)	0.64	0.52	0.62	1.21	0.5	0.70	**0.72 (↑0.08)**
Bottom Area (cm2)	2546.88	2754.68	2466.44	2200	2715.60	2480.12	**2408.96 (↓137.92)**
Height (mm)	570.68	580.65	570.88	343	594.52	553.93	**562.2 (↓8.48)**
Num of Part	54	60.00	54.03	31.04	51.93	57.3	**53.5 (↓0.5)**

Table 6.12 Results of design-relevant parameters in Build 3 by using different optimisation algorithms.

Build 3	Original	GD	GA	DE	SA	BO	PSO
Degree of Part Filling (%)	23.1	21.00	20.54	23	22.42	21.57	**20.83 (↓2.27)**
Part Ratio (WL) (%)	1.23	1.07	1.33	1.36	1.4	1.38	**1.4 (↑0.17)**
Part Ratio (HL) (%)	0.44	0.54	0.55	0.55	0.6	0.58	**0.53 (↑0.09)**
Part Ratio (WH) (%)	2.8	3.05	2.92	3	2.70	2.91	**2.87 (↑0.07)**
Part Height (mm)	40.83	43.66	44.21	40	44.10	40.62	**41.65 (↑0.82)**
Degree of Total Filling (%)	4.69	5.42	3.28	4.76	4.37	4.45	**4.14 (↓0.55)**
Total Ratio (WL) (%)	0.49	0.14	0.45	0.10	0	0.16	**0.42 (↓0.07)**
Total Ratio (HL) (%)	0.17	0.02	0.01	0.20	0	0.11	**0.66 (↑0.49)**
Total Ratio (WH) (%)	2.87	2.47	2.46	2	3.00	2.82	**0.73 (↓2.14)**
Bottom Area (cm2)	1917.71	2124.20	2177.36	2000	2580.92	2848.90	**2029.66 (↑111.95)**
Height (mm)	107	104.53	103.42	103.58	108.43	100.3	**106 (↓1)**
Num of Part	10	10.21	10.63	12.01	12.54	12.35	**12.35 (↑2.35)**

The tables above present the experiment data on optimised design-relevant data and the energy consumption values by using the proposed method. They provide a comparison across three different prototypes of design-relevant parameters on build-level and energy consumption after employing PSO. The main objective is to leverage the proposed method to reduce the unit energy consumption (in Wh/g) and optimise those design-relevant parameters.

In Table 6.10, the PSO method has a part-filling degree of 11.62%, which demonstrates a slight decrease (-0.41) from the original value, but in the third table (Table 6.12), the PSO algorithm has a part-filling degree at 20.83% which is 2.27 lower than the original one. This indicates that PSO fluctuates in the part-filling degree while preserving a good local filling effect in this scenario. In addition, PSO contributes to modifying specifications in terms of coordination of part ratios (width-length, height-length, and width-height). In

Table *6.11*, the part ratio at width-and-length dimension is 1.32, increasing by 1.23 compared to the original values. The same pattern can be observed in Table 6.12, in which the width-and-length ratio reaches 1.4, where a significant improvement is achieved (+0.17). This observation means that PSO has an advantage in the coordination of the width-and-length ratio of the local specification, which has the potential to assist the utilisation efficiency of the design specifications.

Furthermore, PSO has certain modifications in terms of balancing the bottom area and the height. According to

Table *6.11*, the adjustment of the bottom area and the height is employed, with 2628.42 mm^2 (increased by 42.91 mm^2) and 407.75 mm (increased by 36.73 mm), respectively, while a similar pattern can be seen in Table 6.12. These results demonstrate that PSO can support adjusting bottom area coverage and height expansion, potentially achieving a more balanced space utilisation.

Thirdly, the PSO could provide good stability and adaptability of the layout structure. Among the three tables, PSO can help to suggest a relatively stable number of layout segments. In other words, it is helpful to maintain the overall stability and operation of the position for different batches during the working process. In this case study, it is highlighted that PSOs have certain adaptability in different prototypes. For example, PSO performed better in part height and overall filling in Build 1. It also shows merits in part ratios and bottom area in Build 2. Besides, PSO stands out in terms of part ratios and the number of builds.

To sum up, the PSO method contributes to certain optimisation on multiple specifications such as height, overall filling degree, and bottom area. It focuses on layout optimisation, enabling the adjustment in terms of local area. PSO potentially provides control support by optimising part ratios. These proportional relationships may improve the utilisation efficiency and coordination of the workspace. These findings also provide preliminary insights that the combination of the optimised parameters by using PSO and a data-driven approach can effectively reduce the unit energy consumption in these printed objects. These improvements have the potential to enhance the energy efficiency of the builds, providing the usefulness of support and suggestions for part designers and decision-making for process operators. By optimising design-relevant parameters, these designs can present a more sustainable and cost-effective direction to Am systems, strengthening energy-efficient design and in-process sustainability.

6.6 Summary

The study of this section set out to evaluate a hybrid method for predicting energy consumption by using layer-wise images from the sliced CAD model prior to printing and to assess the effectiveness and feasibility of deploying this method on a targeted FPGA platform. The student model on the targeted FPGA platform was integrated with the PSO algorithm to predict the build energy consumption based on the combination of part-design and process-planning parameters. The optimised parameters and unit energy consumption of the build are expected to provide decision support and inform designers and operators before the process.

In the current phase, the proposed method integrating PC and FPGA platforms can collaboratively collect image data from CAD models. By leveraging features from these image data, the predictive model extracts valuable insights from the historical data, which is then trained collaboratively with the PSO algorithm. This allows designers to collect and analyse design-relevant data before the additively manufactured parts start. The generalisability of these results is subject to certain limitations. For instance, the data source is mainly derived from the design aspects, such as geometry, process planning and part design. Due to the complexity of the AM machine, multimodal data from the process and working environment can be collected, including temperature, pressure, audio, gas levels etc.

This approach aims to optimise workflows and promote more cost-effective and sustainable manufacturing processes. In the future, FPGA-CNN can be used to take timely action to predict energy consumption, resulting in significant energy savings and improved operational efficiency. Predictive models provide valuable insights from real-time and historical data, helping to optimise quality control and predictive maintenance in AM processes. Manufacturers can identify and modify process plans for AM machines by monitoring energy consumption patterns in real-time.

Chapter 7 Achievement and Conclusions

7.1 Achievements

The main goal of the current study was to develop energy consumption prediction models for the targeted SLS systems using advanced data-driven techniques such as DL and FPGAs. This framework is expected to contribute to the sustainability and cost-effectiveness of AM systems in a broader context, which is crucial for the future of manufacturing within the framework of I4.0. Through the development and validation of a multi-scale feature fusion model integrated with a robust module, it was found that the energy consumption prediction accuracy and efficiency of the SLS process were improved. This was achieved by utilising image-based features that can significantly influence the energy consumption of each unique layer. A lightweight DL model for predicting energy consumption was developed through research and the use of a KD strategy. The model preserved high predictive performance while optimising computational resources and was therefore well suited for development on FPGAs. In addition, the development of the target FPGA platform speeded up the processing of image data, enabling the provision of features and predicted energy consumption values. This, in turn, facilitated faster feature extraction and energy consumption predictions. These findings supported the optimisation technique, PSO, when integrated with DNN to determine ideal part-design and process-planning parameters, thereby minimising the build energy consumption. This approach would assist part designers and process operators with decision-making and support throughout the design and manufacturing process by offering optimal combinations of design-relevant parameters. Having recalled the research questions presented in Chapter 1, this section will provide the answers to those questions based on the study.

- ***What lightweight deep learning architecture and model compression techniques can be developed to effectively analyse layer-wise image data for energy consumption prediction in SLS when deployed on FPGA platforms?***

Multiple types of data were generated during AM systems, including layer-wise images derived from CAD models, design-relevant data related to part design and process planning, and energy-related data containing energy consumption values for each different layer. To address the challenges associated with handling this variety of data, various methods such as data integration and advanced data-driven approaches (e.g., DL) were utilised. As for the data integration for image data, a multi-scale feature fusion model has been developed, which identifies the features within the layer-wise images extracted from CAD models. Utilising these features, the model could predict the energy consumption of each layer. To achieve lightweight model development, this research employed quantisation and KD, which were adept at handling complex data generated during the AM process. These lightweight models could be developed and accelerated on the FPGA platform. Processing data on the edge device reduces the need to send data to a central server, and the collaboration of FPGA and lightweight models reduces latency and speeds up decision-making while reducing power consumption. In addition, it would have the potential to perform real-time data processing for dynamic tuning during the manufacturing process.

- ***How can the inherent parallel processing and reconfigurability of FPGAs be exploited to enhance the performance and energy efficiency of lightweight neural networks for predictive modelling in AM?***

The second aim of this research was to investigate the unique characteristics of the hardware accelerator, FPGA, which plays an important role in optimising the performance of the lightweight model for predictive modelling in AM scenarios. It is emphasised that the parallel characteristic of the FPGA platform allows the simultaneous execution of multiple operations in DL models, especially for convolutional layers. In addition, FPGAs facilitate the implementation of dedicated algorithms for specific mathematical operations, which is required for the lightweight model. In this study, the convolution operation was achieved in the targeted FPGA

platform. Other advantages including low latency and high throughput, facilitate processing large amounts of data while reducing computational load and power consumption. In order to optimise the lightweight model leveraging these specific characteristics, quantisation techniques further reduce the precision of the original model so that it can be implemented more efficiently on the targeted FPGA. Furthermore, designing an FPGA-compatible architecture of the lightweight model and optimising data flow are required, thereby efficiently allocating resources for the optimisation. These unique strategies contribute to enhancing the performance of the lightweight model in AM predictive modelling.

- ***What are the essential steps and design considerations for integrating an FPGA-based monitoring system for real-time energy consumption analysis in AM, and how does this system enable dynamic optimisation support of energy usage?***

There are several key steps and considerations involved in integrating a monitoring and management system for AM with the collaboration of FPGAs and DL-based approaches. Prior to building the system, energy consumption metrics must be determined and monitoring objectives identified. In this study, unit energy consumption was the metric, and the objective was to determine the optimal parameter combination and minimum energy consumption of the build. In the framework, the first step involved developing the multi-scale feature fusion model serving as a teacher model for predicting energy consumption from layer-wise images based on historical data. The second step leveraged the KD strategy to obtain the lightweight student model. Subsequently, the student model could be deployed and accelerated on the FPGA to process image features and predict energy consumption. This process collaborated with optimisation algorithms to provide optimisation of parameter combinations and minimise the energy consumption of the build based on those parameters. By applying predictive analytics for future energy consumption, the system would alert operators to potential issues before they occur, enabling proactive adjustments on part design and process planning to the AM process.

7.2 Future Works

While this research comprehensively analyses energy consumption predictive modelling in an SLS system, several areas require further investigation to strengthen the basis for future studies. Firstly, a limitation of the current study is the reliance on specific SLS machine configurations and the need for extensive focus on other data in different modalities rather than merely design-relevant data to train energy prediction models, i.e., layer-wise images from CAD models. This problem can be addressed by integrating multimodal data sources such as material properties, machine parameters and environmental settings to enhance the robustness of energy predictive models. Future research can extend the scope by exploring knowledge fusion across data and considering data from different modalities. More advanced data analytics are expected to offer new insights into the predictive modelling process. When considering the deployment of these models, FPGA utilisation contributes to edge computing applications by providing real-time processing due to reducing the computational load and power on central servers.

The second limitation comes from edge computing and FPGA deployment. A compressed model has the potential to affect dynamic tuning by sacrificing the precision of parameters. There is a lack of a more standardised framework for FPGA-accelerated multimodal data processing, making it difficult to guarantee real-time performance while integrating different data streams, where computational overloads must be considered. The high computational load of multimodal fusion on edge devices requires reducing FPGA power consumption while maintaining real-time performance.

Thirdly, the adaptive algorithm for nonlinear dynamics for the SLS system is absent. Therefore, the focus of future work involves predictive lightweight modelling and closed-loop control techniques to provide more accurate and efficient energy

management in SLS systems. It is expected that the targeted system will process data in the real-time environment to provide immediate feedback on inefficiencies or anomalies in energy consumption based on design-relevant parameters. The future study could incorporate the closed-loop control into the SLS system which can significantly optimise the energy management and monitoring system. A closed-loop control system could promote the optimisation of energy management to improve the overall efficiency of the process through continuous monitoring and real-time adjustment of the parameters in the SLS system. For instance, the closed-loop control can be achieved from the following aspects. This approach will increase energy efficiency, improve overall system reliability and performance, and provide the basis for smarter, more sustainable manufacturing processes.

In order to address these challenges, future research direction could include several aspects such as 1) developing more comprehensive models informed by physics, bridging CAD data with physical phenomena and mechanisms, enabling integration of data-driven and physical-constrained methods in the real-time, for example, Physics-Informed Neural Network (PINN), 2) developing a more robust lightweight model to process multimodal data in a more effective and efficient workflow, and 3) designing smarter PID controller for nonlinear SLS dynamics by leveraging reinforcement learning. By doing so, future work can develop more comprehensive, robust and adaptive energy management systems for sustainability practices.

7.3 Conclusions

This study set out to develop a hybrid model for meeting the requirement of energy optimisation in the current SLS system by using different data obtained from real-world scenarios. To begin with, the multi-scale feature fusion model was derived from the enhanced U-Net architecture. By the comparative experiments with prevailing DL models, the proposed model showed the merits of the proposed methodology in feature extraction and fusion from layer-wise image data. This model contributed to the complex feature fusion from the images to predict the energy consumption on each

layer. In order to deploy on the FPGA platform for fast and efficient inference, the KD strategy was applied to obtain the lightweight model. The framework realised the collaboration of DL models and the targeted platform to evaluate the effectiveness and feasibility of the predictive model. In the current stage, the predictive model extracted valuable information from the historical data by leveraging features from the lightweight model, followed by training collaboratively with the optimisation algorithm. This could help part designers and process operators to collect and analyse design-relevant data before the manufacturing process starts. This approach aimed to optimise the design and process parameters and energy consumption, facilitating a more cost-effective and sustainable manufacturing system. In the future, an FPGA-CNN could be employed to take real-time actions to predict energy consumption, potentially leading to significant energy savings and increased operational efficiency. The predictive models would provide valuable insights from both in-situ and historical data, further optimising quality control and predictive maintenance in AM processes. Manufacturers could identify and make corrections to the process plan and designs by monitoring the energy consumption patterns in real-time.

Bibliography

Abdelouahab, K., Pelcat, M., Serot, J. and Berry, F. 2018. Accelerating CNN inference on FPGAs: A Survey.

Ahn, D.-G. 2021. Directed Energy Deposition (DED) Process: State of the Art. *International Journal of Precision Engineering and Manufacturing-Green Technology* 8(2), pp. 703–742. doi: 10.1007/s40684-020-00302-7.

Ahuett-Garza, H. and Kurfess, T. 2018. A brief discussion on the trends of habilitating technologies for Industry 4.0 and Smart manufacturing. *Manufacturing Letters* 15, pp. 60–63. doi: 10.1016/j.mfglet.2018.02.011.

Akhavan, J., Lyu, J. and Manoochehri, S. 2024. A deep learning solution for real-time quality assessment and control in additive manufacturing using point cloud data. *Journal of Intelligent Manufacturing* 35(3), pp. 1389–1406. doi: 10.1007/s10845-023-02121-4.

Akilan, I. and Velmurugan, C. 2022. *Mechanical Testing of Additive Manufacturing Materials*. Springer International Publishing. Available at: http://dx.doi.org/10.1007/978-3-030-89401-6_11.

Albakri, M.I., Sturm, L.D., Williams, C.B. and Tarazaga, P.A. 2017. Impedance-based non-destructive evaluation of additively manufactured parts. *Rapid Prototyping Journal* 23(3), pp. 589–601. doi: 10.1108/RPJ-03-2016-0046.

Albawi, S., Mohammed, T.A. and Al-Zawi, S. 2017. Understanding of a convolutional neural network. In: *2017 International Conference on Engineering and Technology (ICET)*. IEEE, pp. 1–6. doi: 10.1109/ICEngTechnol.2017.8308186.

ALMASRI, W., DANGLADE, F., BETTEBGHOR, D., ADJED, F. and ABABSA, F. 2022. Deep Learning for Additive Manufacturing-driven Topology Optimization. *Procedia CIRP* 109(March), pp. 49–54. doi: 10.1016/j.procir.2022.05.317.

Alzubaidi, L. et al. 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data* 8(1), p. 53. doi: 10.1186/s40537-021-00444-8.

Ba, L.J. and Caruana, R. 2013. Do Deep Nets Really Need to be Deep? In: *Proceedings of the 27th International Conference on Neural Information Processing Systems*. Montreal, Canada: MIT Press, pp. 2654–2662.

Babu, S.S., Mourad, A.-H.I., Harib, K.H. and Vijayavenkataraman, S. 2023. Recent developments in the application of machine-learning towards accelerated predictive multiscale design and additive manufacturing. *Virtual and Physical Prototyping* 18(1). doi: 10.1080/17452759.2022.2141653.

Bahnini, I., Rivette, M., Rechia, A., Siadat, A. and Elmesbahi, A. 2018. Additive manufacturing technology: the status, applications, and prospects. *The International Journal of Advanced Manufacturing Technology* 97(1–4), pp. 147–161. Available at: http://link.springer.com/10.1007/s00170-018-1932-y.

Baumers, M., Tuck, C., Bourell, D.L., Sreenivasan, R. and Hague, R. 2011. Sustainability of additive manufacturing: measuring the energy consumption of the laser sintering process. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 225(12), pp. 2228–2239. doi: 10.1177/0954405411406044.

Baumers, M., Tuck, C., Wildman, R., Ashcroft, I., Rosamond, E. and Hague, R. 2013. Transparency Built-in. *Journal of Industrial Ecology* 17(3), pp. 418–431. doi: 10.1111/j.1530-9290.2012.00512.x.

Biookaghazadeh, S., Zhao, M. and Ren, F. 2018. Are FPGAs suitable for edge computing?

Birant, D. and Kut, A. 2007. ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering* 60(1), pp. 208–221. doi: 10.1016/j.datak.2006.01.013.

Bisht, P.S. and Awasthi, A. 2020. Design and Analysis of Composite and Al Alloy Wheel Rim. In: *Lecture Notes on Multidisciplinary Industrial Engineering*. pp. 15–29. doi: 10.1007/978-981-15-4331-9_2.

Boutros, A. and Betz, V. 2021. FPGA Architecture: Principles and Progression. *IEEE Circuits and Systems Magazine* 21(2), pp. 4–29. doi: 10.1109/MCAS.2021.3071607.

Braconnier, D.J., Jensen, R.E. and Peterson, A.M. 2020. Processing parameter correlations in material extrusion additive manufacturing. *Additive Manufacturing* 31(January 2019), p. 100924. doi: 10.1016/j.addma.2019.100924.

Buciluǎ, C., Caruana, R. and Niculescu-Mizil, A. 2006. Model compression. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, pp. 535–541. Available at: https://dl.acm.org/doi/10.1145/1150402.1150464.

Campbell, R.I., Jee, H. and Kim, Y.S. 2013. Adding product value through additive manufacturing. *Proceedings of the International Conference on Engineering Design, ICED* 4 DS75-04(August), pp. 259–268.

Chandra Sugianto, W. and Soo Kim, B. 2024. Particle swarm optimization for integrated scheduling problem with batch additive manufacturing and batch direct-shipping delivery. *Computers & Operations Research* 161(August 2023), p. 106430. doi: 10.1016/j.cor.2023.106430.

Chaunier, L., Guessasma, S., Belhabib, S., Della Valle, G., Lourdin, D. and Leroy, E. 2018. Material extrusion of plant biopolymers: Opportunities & challenges for 3D printing. *Additive Manufacturing* 21(March), pp. 220–233. doi: 10.1016/j.addma.2018.03.016.

Chen, H., Wang, Y., Xu, C., Xu, C. and Tao, D. 2021. Learning Student Networks via Feature Embedding. *IEEE Transactions on Neural Networks and Learning Systems* 32(1), pp. 25–35. Available at: https://ieeexplore.ieee.org/document/9007474/.

Chen, T. and Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, pp. 785–794. doi: 10.1145/2939672.2939785.

Cheng, J., Wang, P., Li, G., Hu, Q. and Lu, H. 2018. Recent advances in efficient computation of deep convolutional neural networks. *Frontiers of Information Technology & Electronic Engineering* 19(1), pp. 64–77. doi: 10.1631/FITEE.1700789.

Cheng, Y., Wang, D., Zhou, P. and Zhang, T. 2017. A Survey of Model Compression and Acceleration for Deep Neural Networks. pp. 1–10.

Chinchanikar, S. and Shaikh, A.A. 2022. A Review on Machine Learning, Big Data Analytics, and Design for Additive Manufacturing for Aerospace Applications. *Journal of Materials Engineering and Performance* 31(8), pp. 6112–6130. doi: 10.1007/s11665-022-07125-4.

Cho, J.H. and Hariharan, B. 2019. On the Efficacy of Knowledge Distillation. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, pp. 4793–4801. doi: 10.1109/ICCV.2019.00489.

Choudhary, T., Mishra, V., Goswami, A. and Sarangapani, J. 2020. *A comprehensive survey on model compression and acceleration*. Springer Netherlands. doi: 10.1007/s10462-020-09816-7.

Cong, J. and Xiao, B. 2014. Minimizing Computation in Convolutional Neural Networks. In: Wermter, S. et al. eds. *Artificial Neural Networks and Machine Learning -- ICANN 2014*. Cham: Springer International Publishing, pp. 281–290.

Davoudinejad, A. 2021. *Vat photopolymerization methods in additive manufacturing*. Elsevier Inc. Available at: http://dx.doi.org/10.1016/B978-0-12-818411-0.00007-0.

Dev Singh, D., Mahender, T. and Raji Reddy, A. 2021. Powder bed fusion process: A brief review. *Materials Today: Proceedings* 46, pp. 350–355. doi: 10.1016/j.matpr.2020.08.415.

Devesse, W., De Baere, D., Hinderdael, M. and Guillaume, P. 2016. Hardware-in-the-loop control of additive manufacturing processes using temperature feedback. *Journal of Laser Applications* 28(2). doi: 10.2351/1.4943911.

Ding, X., Guo, Y., Ding, G. and Han, J. 2019. ACNet: Strengthening the Kernel Skeletons for Powerful CNN via Asymmetric Convolution Blocks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* 2019-Octob, pp. 1911–1920. doi: 10.1109/ICCV.2019.00200.

Draper, B.A., Beveridge, J.R., Bohm, A.P.W., Ross, C. and Chawathe, M. 2003. Accelerated image processing on FPGAs. *IEEE Transactions on Image Processing* 12(12), pp. 1543–1551. doi: 10.1109/TIP.2003.819226.

Dunaway, D., Harstvedt, J.D. and Ma, J. 2017. A preliminary experimental study of additive manufacturing energy consumption. *Proceedings of the ASME Design Engineering Technical Conference* 4, pp. 1–8. doi: 10.1115/DETC2017-67864.

Dziugaite, G.K. and Roy, D.M. 2015. Neural Network Matrix Factorization. pp. 1–7.

Ekerer, S.C., Boğa, C., Seyedzavvar, M., Koroglu, T. and Farsadi, T. 2024. Optimizing parameters for additive manufacturing: a study on the vibrational performance of 3D printed cantilever beams using material extrusion. *Rapid Prototyping Journal* (September). doi: 10.1108/RPJ-03-2024-0146.

El-Rashidy, N., El-Sappagh, S., Abuhmed, T., Abdelrazek, S. and El-Bakry, H.M. 2020. Intensive Care Unit Mortality Prediction: An Improved Patient-Specific Stacking Ensemble Model. *IEEE Access* 8, pp. 133541–133564. doi: 10.1109/ACCESS.2020.3010556.

Elkaseer, A., Chen, K.J., Janhsen, J.C., Refle, O., Hagenmeyer, V. and Scholz, S.G. 2022. Material jetting for advanced applications: A state-of-the-art review, gaps and future directions. *Additive Manufacturing* 60(PA), p. 103270. doi: 10.1016/j.addma.2022.103270.

Erps, T. et al. 2021. Accelerated discovery of 3D printing materials using data-driven multiobjective optimization. *Science Advances* 7(42), pp. 1–10. doi: 10.1126/sciadv.abf7435.

Ester, M., Kriegel, H.P., Sander, J. and Xiaowei, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*. Portland, Oregon: AAAI Press, pp. 226–231.

Fang, J. et al. 2024. A New Particle Swarm Optimization Algorithm for Outlier Detection: Industrial Data Clustering in Wire Arc Additive Manufacturing. *IEEE Transactions on Automation Science and Engineering* 21(2), pp. 1244–1257. doi: 10.1109/TASE.2022.3230080.

Fischer, F.G., Zimmermann, M.G., Praetzsch, N. and Knaak, C. 2022. Monitoring of the powder bed quality in metal additive manufacturing using deep transfer learning.

Materials and Design 222, p. 111029. Available at: https://doi.org/10.1016/j.matdes.2022.111029.

Frazier, W.E. 2014. Metal additive manufacturing: A review. *Journal of Materials Engineering and Performance* 23(6), pp. 1917–1928. doi: 10.1007/s11665-014-0958-z.

Freitas, D., Almeida, H.A., Bártolo, H. and Bártolo, P.J. 2016. Sustainability in extrusion-based additive manufacturing technologies. *Progress in Additive Manufacturing* 1(1–2), pp. 65–78. doi: 10.1007/s40964-016-0007-6.

Fu, Y., Downey, A.R.J., Yuan, L., Zhang, T., Pratt, A. and Balogun, Y. 2022. Machine learning algorithms for defect detection in metal laser-based additive manufacturing: A review. *Journal of Manufacturing Processes* 75(December 2021), pp. 693–710. doi: 10.1016/j.jmapro.2021.12.061.

Fukuda, T., Suzuki, M., Kurata, G., Thomas, S., Cui, J. and Ramabhadran, B. 2017. Efficient Knowledge Distillation from an Ensemble of Teachers. In: *Interspeech 2017*. ISCA: ISCA, pp. 3697–3701. doi: 10.21437/Interspeech.2017-614.

Fulga, S., Davidescu, A. and Effenberger, I. 2017. Identification of in-line defects and failures during Additive Manufacturing Powder Bed Fusion processes. Oancea, G. and Drăgoi, M. V. eds. *MATEC Web of Conferences* 94, p. 03005. doi: 10.1051/matecconf/20179403005.

Gandhare, S. and Karthikeyan, B. 2019. Survey on FPGA Architecture and Recent Applications. *Proceedings - International Conference on Vision Towards Emerging Trends in Communication and Networking, ViTECoN 2019*, pp. 1–4. doi: 10.1109/ViTECoN.2019.8899550.

Gao, M., Li, L., Wang, Q., Liu, C., Li, X. and Liu, Z. 2024. Feature-based energy consumption quantitation strategy for complex additive manufacturing parts. *Energy* 297(July 2023), p. 131249. doi: 10.1016/j.energy.2024.131249.

Ghansiyal, S., Yi, L., Steiner-Stark, J., Müller, M.M., Kirsch, B., Glatt, M. and Aurich, J.C. 2023. A conceptual framework for layerwise energy prediction in laser-based powder bed fusion process using machine learning. *Procedia CIRP* 116, pp. 7–12. doi: 10.1016/j.procir.2023.02.002.

Gheisari, M., Wang, G. and Bhuiyan, M.Z.A. 2017. A Survey on Deep Learning in Big Data. *Proceedings - 2017 IEEE International Conference on Computational Science and Engineering and IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, CSE and EUC 2017* 2, pp. 173–180. doi: 10.1109/CSE-EUC.2017.215.

Ghungrad, S. and Haghighi, A. 2024. Kinematics-guided data-driven energy surrogate model for robotic additive manufacturing. *Manufacturing Letters* 41, pp. 133–142. doi: 10.1016/j.mfglet.2024.09.017.

Gibson, I., Rosen, D., Stucker, B. and Khorasani, M. 2021. *Additive Manufacturing Technologies*. Cham: Springer International Publishing. doi: 10.1007/978-3-030-56127-7.

Goh, G.D., Yap, Y.L., Tan, H.K.J., Sing, S.L., Goh, G.L. and Yeong, W.Y. 2020. Process–Structure–Properties in Polymer Additive Manufacturing via Material Extrusion: A Review. *Critical Reviews in Solid State and Materials Sciences* 45(2), pp. 113–133. doi: 10.1080/10408436.2018.1549977.

Goodfellow, I., Bengio, Y. and Courville, A. 2016. *Deep Learning*. The MIT Press.

Gou, J., Yu, B., Maybank, S.J. and Tao, D. 2021. Knowledge Distillation: A Survey. *International Journal of Computer Vision* 129(6), pp. 1789–1819. doi: 10.1007/s11263-021-01453-z.

Gu, J. et al. 2018. Recent advances in convolutional neural networks. *Pattern Recognition* 77, pp. 354–377. doi: 10.1016/j.patcog.2017.10.013.

Gülcan, O., Günaydın, K. and Tamer, A. 2021. The State of the Art of Material Jetting—A Critical Review. *Polymers* 13(16), p. 2829. doi: 10.3390/polym13162829.

Guo, K., Sui, L., Qiu, J., Yao, S., Han, S., Wang, Y. and Yang, H. 2016. Angel-eye: A complete design flow for mapping CNN onto customized hardware. *Proceedings of IEEE Computer Society Annual Symposium on VLSI, ISVLSI* 2016-Septe, pp. 24–29. doi: 10.1109/ISVLSI.2016.129.

Gutierrez-Osorio, A.H., Ruiz-Huerta, L., Caballero-Ruiz, A., Siller, H.R. and Borja, V. 2019. Energy consumption analysis for additive manufacturing processes. *The*

International Journal of Advanced Manufacturing Technology 105(1–4), pp. 1735–1743. doi: 10.1007/s00170-019-04409-3.

Guzel Aydin, S. and Bilge, H.S. 2021. FPGA -Based Implementation of Convolutional Layer Accelerator Part for CNN. In: *2021 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, pp. 1–6. doi: 10.1109/ASYU52992.2021.9599029.

Hague, R., Campbell, I. and Dickens, P. 2003. Implications on design of rapid manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 217(1), pp. 25–30. doi: 10.1243/095440603762554587.

Haleem, A. and Javaid, M. 2019. Additive Manufacturing Applications in Industry 4.0: A Review. *Journal of Industrial Integration and Management* 04(04), p. 1930001. doi: 10.1142/S2424862219300011.

Han, S., Pool, J., Tran, J. and Dally, W.J. 2015. Learning both weights and connections for efficient neural networks. *Advances in Neural Information Processing Systems* 2015-Janua, pp. 1135–1143.

Han, W., Kong, L. and Xu, M. 2022. Advances in selective laser sintering of polymers. *International Journal of Extreme Manufacturing* 4(4). doi: 10.1088/2631-7990/ac9096.

Hasan, M.R., Liu, Z. and Rahman, A. 2023. Energy Consumption Modeling for DED-based Hybrid Additive Manufacturing. *The International Journal of Advanced Manufacturing Technology*, pp. 1–19.

He, K., Zhang, X., Ren, S. and Sun, J. 2014. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. pp. 346–361. doi: 10.1007/978-3-319-10578-9_23.

Hinton, G., Vinyals, O. and Dean, J. 2015. Distilling the Knowledge in a Neural Network. pp. 1–9.

Hinton, G.E., Osindero, S. and Teh, Y.-W. 2006. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation* 18(7), pp. 1527–1554. doi: 10.1162/neco.2006.18.7.1527.

Horn, T.J. and Harrysson, O.L.A. 2012. Overview of current additive manufacturing technologies and selected applications. *Science Progress* 95(3), pp. 255–282. doi: 10.3184/003685012X13420984463047.

Hu, F., Qin, J., Li, Y., Liu, Y. and Sun, X. 2021a. Deep Fusion for Energy Consumption Prediction in Additive Manufacturing. *Procedia CIRP* 104(March), pp. 1878–1883. doi: 10.1016/j.procir.2021.11.317.

Hu, L., Wang, Y., Shu, L., Cai, W., Lv, J. and Xu, K. 2023. Energy benchmark for evaluating the energy efficiency of selective laser melting processes. *Applied Thermal Engineering* 221, p. 119870. Available at: https://www.sciencedirect.com/science/article/pii/S1359431122018002.

Hu, X., Chu, L., Pei, J., Liu, W. and Bian, J. 2021b. Model complexity of deep learning: a survey. *Knowledge and Information Systems* 63(10), pp. 2585–2619. doi: 10.1007/s10115-021-01605-0.

Huang, J., Chen, Q., Jiang, H., Zou, B., Li, L., Liu, J. and Yu, H. 2020. A survey of design methods for material extrusion polymer 3D printing. *Virtual and Physical Prototyping* 15(2), pp. 148–162. Available at: https://www.tandfonline.com/doi/full/10.1080/17452759.2019.1708027.

Huang, Z. and Wang, N. 2017. Like What You Like: Knowledge Distill via Neuron Selectivity Transfer. doi: 10.48550/arXiv.1707.01219.

ISO/ASTM. 2013. *Additive Manufacturing - General Principles Terminology (ASTM52900)*. doi: 10.1520/F2792-12A.2.

Jacob, B. et al. 2018. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2704–2713. doi: 10.1109/CVPR.2018.00286.

Janiesch, C., Zschech, P. and Heinrich, K. 2021. Machine learning and deep learning. *Electronic Markets* 31(3), pp. 685–695. doi: 10.1007/s12525-021-00475-2.

Ji, M., Peng, G., Li, S., Cheng, F., Chen, Z., Li, Z. and Du, H. 2022. A neural network compression method based on knowledge-distillation and parameter quantization for the bearing fault diagnosis. *Applied Soft Computing* 127, p. 109331. doi: 10.1016/j.asoc.2022.109331.

Jiang, H., Li, J., Yi, S., Wang, X. and Hu, X. 2011. A new hybrid method based on partitioning-based DBSCAN and ant clustering. *Expert Systems with Applications* 38(8), pp. 9373–9381. doi: 10.1016/j.eswa.2011.01.135.

Jiang, J., Xiong, Y., Zhang, Z. and Rosen, D.W. 2022. Machine learning integrated design for additive manufacturing. *Journal of Intelligent Manufacturing* 33(4), pp. 1073–1086. doi: 10.1007/s10845-020-01715-6.

Jiménez, M., Romero, L., Domínguez, I.A., Espinosa, M.D.M. and Domínguez, M. 2019. Additive Manufacturing Technologies: An Overview about 3D Printing Methods and Future Prospects. García-Alcaraz, J. L. ed. *Complexity* 2019(1). Available at: https://onlinelibrary.wiley.com/doi/10.1155/2019/9656938.

Jin, Z., Zhang, Z., Demir, K. and Gu, G.X. 2020. Machine Learning for Advanced Additive Manufacturing. *Matter* 3(5), pp. 1541–1556. doi: 10.1016/j.matt.2020.08.023.

Jing Yang, Xiaoqin Zeng, Shuiming Zhong and Shengli Wu. 2013. Effective Neural Network Ensemble Approach for Improving Generalization Performance. *IEEE Transactions on Neural Networks and Learning Systems* 24(6), pp. 878–887. doi: 10.1109/TNNLS.2013.2246578.

Johnson, N.S., Vulimiri, P.S., To, A.C., Zhang, X., Brice, C.A., Kappes, B.B. and Stebner, A.P. 2020. Invited review: Machine learning for materials developments in metals additive manufacturing. *Additive Manufacturing* 36. doi: 10.1016/j.addma.2020.101641.

Kaikai, X., Yadong, G. and Qiang, Z. 2023. Comparison of traditional processing and additive manufacturing technologies in various performance aspects: a review.

Archives of Civil and Mechanical Engineering 23(3), pp. 1–28. doi: 10.1007/s43452-023-00699-3.

Kanishka, K. and Acherjee, B. 2023. Revolutionizing manufacturing: A comprehensive overview of additive manufacturing processes, materials, developments, and challenges. *Journal of Manufacturing Processes* 107(September), pp. 574–619. Available at: https://doi.org/10.1016/j.jmapro.2023.10.024.

Kellens, K., Baumers, M., Gutowski, T.G., Flanagan, W., Lifset, R. and Duflou, J.R. 2017. Environmental Dimensions of Additive Manufacturing: Mapping Application Domains and Their Environmental Implications. *Journal of Industrial Ecology* 21(S1), pp. S49–S68. doi: 10.1111/jiec.12629.

Kellens, K., Renaldi, R., Dewulf, W., Kruth, J.P. and Duflou, J.R. 2014. Environmental impact modeling of selective laser sintering processes. *Rapid Prototyping Journal* 20(6), pp. 459–470. doi: 10.1108/RPJ-02-2013-0018.

Kellens, K., Yasa, E., Renaldi, R., Dewulf, W., Kruth, J.-P. and Duflou, J. 2011. Energy and Resource Efficiency of SLS/SLM Processes (Keynote Paper). In: *SFF Symposium 2011*. pp. 1–16.

Khorram Niaki, M. and Nonino, F. 2017. Additive manufacturing management: a review and future research agenda. *International Journal of Production Research* 55(5), pp. 1419–1439. doi: 10.1080/00207543.2016.1229064.

Kumar, D., Liu, Y., Song, H. and Namilae, S. 2024. Explainable deep neural network for in-plain defect detection during additive manufacturing. *Rapid Prototyping Journal* 30(1), pp. 49–59. doi: 10.1108/RPJ-05-2023-0157.

Kumar, N.S. and Madhumati, G.L. 2023. Implementation of Convolutional Neural Networks on FPGA for Object Detection. In: *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, pp. 1–5. doi: 10.1109/ICCCNT56998.2023.10307606.

Kumar, S. 2003. Selective laser sintering: A qualitative and objective approach. *JOM* 55(10), pp. 43–47. doi: 10.1007/s11837-003-0175-y.

Leary, M., Merli, L., Torti, F., Mazur, M. and Brandt, M. 2014. Optimal topology for additive manufacture: A method for enabling additive manufacture of support-free optimal structures. *Materials & Design* 63, pp. 678–690. doi: 10.1016/j.matdes.2014.06.015.

LeCun, Y. 2019. 1.1 Deep Learning Hardware: Past, Present, and Future. In: *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*. IEEE, pp. 12–19. doi: 10.1109/ISSCC.2019.8662396.

LeCun, Y., Bengio, Y. and Hinton, G. 2015. Deep learning. *Nature* 521(7553), pp. 436–444. doi: 10.1038/nature14539.

Li, M., Halstead, M. and Mccool, C. 2024. Knowledge Distillation for Efficient Instance Semantic Segmentation with Transformers. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern RecognitionProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshop*, pp. 5432–5439.

Li, W., Zhang, H., Wang, G., Xiong, G., Zhao, M., Li, G. and Li, R. 2023. Deep learning based online metallic surface defect detection method for wire and arc additive manufacturing. *Robotics and Computer-Integrated Manufacturing* 80(September 2022), p. 102470. doi: 10.1016/j.rcim.2022.102470.

Li, X., Jia, X., Yang, Q. and Lee, J. 2020. Quality analysis in metal additive manufacturing with deep learning. *Journal of Intelligent Manufacturing* 31(8), pp. 2003–2017. Available at: https://doi.org/10.1007/s10845-020-01549-2.

Li, Z., Liu, F., Yang, W., Peng, S. and Zhou, J. 2022. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems* 33(12), pp. 6999–7019. doi: 10.1109/TNNLS.2021.3084827.

Lim, J.-S., Oh, W.-J., Lee, C.-M. and Kim, D.-H. 2021. Selection of effective manufacturing conditions for directed energy deposition process using machine learning methods. *Scientific Reports* 11(1), p. 24169. doi: 10.1038/s41598-021-03622-z.

Lin, J., Chen, W.-M., Lin, Y., Cohn, J., Gan, C. and Han, S. 2020. MCUNet: Tiny Deep Learning on IoT Devices. In: *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*. Vancouver, pp. 1–15.

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B. and Belongie, S. 2017. Feature Pyramid Networks for Object Detection. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 936–944. doi: 10.1109/CVPR.2017.106.

Liu, B., Long, R. and Chou, K.C. 2016. IDHS-EL: Identifying DNase i hypersensitive sites by fusing three different modes of pseudo nucleotide composition into an ensemble learning framework. *Bioinformatics* 32(16), pp. 2411–2418. doi: 10.1093/bioinformatics/btw186.

Liu, C., Wang, R.R., Ho, I., Kong, Z.J., Williams, C., Babu, S. and Joslin, C. 2023. Toward online layer-wise surface morphology measurement in additive manufacturing using a deep learning-based approach. *Journal of Intelligent Manufacturing* 34(6), pp. 2673–2689. doi: 10.1007/s10845-022-01933-0.

Liu, W., Deng, K., Wei, H., Zhao, P., Li, J. and Zhang, Y. 2021a. A decision-making model for comparing the energy demand of additive-subtractive hybrid manufacturing and conventional subtractive manufacturing based on life cycle method. *Journal of Cleaner Production* 311(April), p. 127795. doi: 10.1016/j.jclepro.2021.127795.

Liu, Y., Li, J., Cheng, T., Fan, Z., Li, W., Xia, W. and Wei, Q. 2024. Parameter automatic optimization strategy for laser powder bed fusion using neural network infrared radiation intensity prediction model. *Additive Manufacturing* 92(April), p. 104373. doi: 10.1016/j.addma.2024.104373.

Liu, Z., He, B., Lyu, T. and Zou, Y. 2021b. A Review on Additive Manufacturing of Titanium Alloys for Aerospace Applications: Directed Energy Deposition and Beyond Ti-6Al-4V. *JOM* 73(6), pp. 1804–1818. doi: 10.1007/s11837-021-04670-6.

Liu, Z., Sun, M., Zhou, T., Huang, G. and Darrell, T. 2018a. Rethinking the Value of Network Pruning. *7th International Conference on Learning Representations, ICLR 2019*, pp. 1–21.

Liu, Z.Y., Li, C., Fang, X.Y. and Guo, Y.B. 2018b. Energy Consumption in Additive Manufacturing of Metal Parts. *Procedia Manufacturing* 26, pp. 834–845. doi: 10.1016/j.promfg.2018.07.104.

Lores, A., Azurmendi, N., Agote, I. and Zuza, E. 2019. A review on recent developments in binder jetting metal additive manufacturing: materials and process characteristics. *Powder Metallurgy* 62(5), pp. 267–296. doi: 10.1080/00325899.2019.1669299.

Lu, L., Hou, J., Yuan, S., Yao, X., Li, Y. and Zhu, J. 2023a. Deep learning-assisted real-time defect detection and closed-loop adjustment for additive manufacturing of continuous fiber-reinforced polymer composites. *Robotics and Computer-Integrated Manufacturing* 79(April 2022), p. 102431. doi: 10.1016/j.rcim.2022.102431.

Lu, W., Lee, N.A. and Buehler, M.J. 2023b. Modeling and design of heterogeneous hierarchical bioinspired spider web structures using deep learning and additive manufacturing. *Proceedings of the National Academy of Sciences* 120(31), p. 2017. doi: 10.1073/pnas.2305273120.

Luo, Y. and Chen, Y. 2021. FPGA-Based Acceleration on Additive Manufacturing Defects Inspection. *Sensors* 21(6), p. 2123. doi: 10.3390/s21062123.

Lv, X., Ye, F., Cheng, L., Fan, S. and Liu, Y. 2019. Binder jetting of ceramics: Powders, binders, printing parameters, equipment, and post-treatment. *Ceramics International* 45(10), pp. 12609–12624. doi: 10.1016/j.ceramint.2019.04.012.

Ma, F., Zhang, H., Hon, K.K.B. and Gong, Q. 2018. An optimization approach of selective laser sintering considering energy consumption and material cost. *Journal of Cleaner Production* 199, pp. 529–537. Available at: https://doi.org/10.1016/j.jclepro.2018.07.185.

Ma, Z., Gao, M., Wang, Q., Wang, N., Li, L., Liu, C. and Liu, Z. 2021. Energy consumption distribution and optimization of additive manufacturing. *International Journal of Advanced Manufacturing Technology* 116(11–12), pp. 3377–3390. doi: 10.1007/s00170-021-07653-8.

Magyari, A. and Chen, Y. 2022. Review of State-of-the-Art FPGA Applications in IoT Networks. *Sensors* 22(19), p. 7496. doi: 10.3390/s22197496.

Majeed, A., Zhang, Y., Ren, S., Lv, J., Peng, T., Waqar, S. and Yin, E. 2021. A big data-driven framework for sustainable and smart additive manufacturing. *Robotics and Computer-Integrated Manufacturing* 67(July 2020), p. 102026. doi: 10.1016/j.rcim.2020.102026.

Manivannan, S. 2023. Automatic quality inspection in additive manufacturing using semi-supervised deep learning. *Journal of Intelligent Manufacturing* 34(7), pp. 3091–3108. Available at: https://doi.org/10.1007/s10845-022-02000-4.

Mies, D., Marsden, W. and Warde, S. 2016. Overview of Additive Manufacturing Informatics: "A Digital Thread." *Integrating Materials and Manufacturing Innovation* 5(1), pp. 114–142. doi: 10.1186/s40192-016-0050-7.

Minar, M.R. and Naher, J. 2018. Recent Advances in Deep Learning: An Overview. 2006, pp. 1–31. doi: 10.13140/RG.2.2.24831.10403.

Mirzadeh, S.I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A. and Ghasemzadeh, H. 2020. Improved Knowledge Distillation via Teacher Assistant. *Proceedings of the AAAI Conference on Artificial Intelligence* 34(04), pp. 5191–5198. doi: 10.1609/aaai.v34i04.5963.

Mittal, S. 2020. A survey of FPGA-based accelerators for convolutional neural networks. *Neural Computing and Applications* 32(4), pp. 1109–1139. doi: 10.1007/s00521-018-3761-1.

Mostafaei, A. et al. 2021. Binder jet 3D printing—Process parameters, materials, properties, modeling, and challenges. *Progress in Materials Science* 119(June 2020), p. 100707. doi: 10.1016/j.pmatsci.2020.100707.

Mueller, R., Teubner, J. and Alonso, G. 2009. Data processing on FPGAs. *Proceedings of the VLDB Endowment* 2(1), pp. 910–921. doi: 10.14778/1687627.1687730.

Murat, F., Kaymaz, İ., Şensoy, A.T. and Korkmaz, İ.H. 2023. Determining the Optimum Process Parameters of Selective Laser Melting via Particle Swarm Optimization Based on the Response Surface Method. *Metals and Materials International* 29(1), pp. 59–70. doi: 10.1007/s12540-022-01205-9.

Ng, W.L., Goh, G.L., Goh, G.D., Ten, J.S.J. and Yeong, W.Y. 2024. Progress and Opportunities for Machine Learning in Materials and Processes of Additive Manufacturing. *Advanced Materials* 36(34). doi: 10.1002/adma.202310006.

Niaki, M.K., Torabi, S.A. and Nonino, F. 2019. Why manufacturers adopt additive manufacturing technologies: The role of sustainability. *Journal of Cleaner Production* 222, pp. 381–392. doi: 10.1016/j.jclepro.2019.03.019.

Nohut, S. and Schwentenwein, M. 2024. Machine learning assisted material development for lithography-based additive manufacturing of porous alumina ceramics. *Open Ceramics* 18(January), p. 100573. Available at: https://doi.org/10.1016/j.oceram.2024.100573.

Obikawa, T., Yoshino, M. and Shinozuka, J. 1999. Sheet steel lamination for rapid manufacturing. *Journal of Materials Processing Technology* 89–90, pp. 171–176. doi: 10.1016/S0924-0136(99)00027-8.

Pagac, M., Hajnys, J., Ma, Q.-P., Jancar, L., Jansa, J., Stefek, P. and Mesicek, J. 2021. A Review of Vat Photopolymerization Technology: Materials, Applications, Challenges, and Future Trends of 3D Printing. *Polymers* 13(4), p. 598. doi: 10.3390/polym13040598.

Papadimitriou, I., Gialampoukidis, I., Vrochidis, S. and Kompatsiaris, I. 2024. AI methods in materials design, discovery and manufacturing: A review. *Computational Materials Science* 235(January), p. 112793. Available at: https://doi.org/10.1016/j.commatsci.2024.112793.

Park, J., Tari, M.J. and Hahn, H.T. 2000. Characterization of the laminated object manufacturing (LOM) process. *Rapid Prototyping Journal* 6(1), pp. 36–50. doi: 10.1108/13552540010309868.

Park, W., Kim, D., Lu, Y. and Cho, M. 2019. Relational Knowledge Distillation. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 3962–3971. doi: 10.1109/CVPR.2019.00409.

Pascanu, R., Gulcehre, C., Cho, K. and Bengio, Y. 2014. How to construct deep recurrent neural networks. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, pp. 1–13.

Passalis, N., Tzelepi, M. and Tefas, A. 2021. Probabilistic Knowledge Transfer for Lightweight Deep Representation Learning. *IEEE Transactions on Neural Networks and Learning Systems* 32(5), pp. 2030–2039. doi: 10.1109/TNNLS.2020.2995884.

Pathak, V.K. and Singh, A.K. 2017. A Particle Swarm Optimization Approach for Minimizing GD&T Error in Additive Manufactured Parts. *International Journal of Manufacturing, Materials, and Mechanical Engineering* 7(3), pp. 69–80. doi: 10.4018/IJMMME.2017070104.

Paul, R. and Anand, S. 2012. Process energy analysis and optimization in selective laser sintering. *Journal of Manufacturing Systems* 31(4), pp. 429–437. doi: 10.1016/j.jmsy.2012.07.004.

Peng, T., Kellens, K., Tang, R., Chen, C. and Chen, G. 2018. Sustainability of additive manufacturing: An overview on its energy demand and environmental impact. *Additive Manufacturing* 21(June 2017), pp. 694–704. doi: 10.1016/j.addma.2018.04.022.

Pereira, T., Kennedy, J. V. and Potgieter, J. 2019. A comparison of traditional manufacturing vs additive manufacturing, the best method for the job. *Procedia Manufacturing* 30, pp. 11–18. doi: 10.1016/j.promfg.2019.02.003.

Pérez, I. and Figueroa, M. 2021. A heterogeneous hardware accelerator for image classification in embedded systems. *Sensors* 21(8). doi: 10.3390/s21082637.

Pérez, M., Carou, D., Rubio, E.M. and Teti, R. 2020. Current advances in additive manufacturing. *Procedia CIRP* 88(March), pp. 439–444. doi: 10.1016/j.procir.2020.05.076.

Pham, T.Q.D. et al. 2023. Fast and accurate prediction of temperature evolutions in additive manufacturing process using deep learning. *Journal of Intelligent Manufacturing* 34(4), pp. 1701–1719. doi: 10.1007/s10845-021-01896-8.

Philip, N.M. and Sivamangai, N.M. 2022. Review of FPGA-Based Accelerators of Deep Convolutional Neural Networks. *ICDCS 2022 - 2022 6th International Conference on Devices, Circuits and Systems* (April), pp. 183–189. doi: 10.1109/ICDCS54290.2022.9780689.

Phuong, M. and Lampert, C. 2019. Distillation-Based Training for Multi-Exit Architectures. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, pp. 1355–1364. doi: 10.1109/ICCV.2019.00144.

Polino, A., Pascanu, R. and Alistarh, D. 2018. Model compression via distillation and quantization. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings* (2015), pp. 1–21.

Prashar, G., Vasudev, H. and Bhuddhi, D. 2023. Additive manufacturing: expanding 3D printing horizon in industry 4.0. *International Journal on Interactive Design and Manufacturing (IJIDeM)* 17(5), pp. 2221–2235. doi: 10.1007/s12008-022-00956-4.

Qiao, Y., Shen, J., Xiao, T., Yang, Q., Wen, M. and Zhang, C. 2017. FPGA-accelerated deep convolutional neural networks for high throughput and energy efficiency. *Concurrency and Computation: Practice and Experience* 29(20), p. e3850. doi: 10.1002/cpe.3850.

Qin, J. et al. 2022. Research and application of machine learning for additive manufacturing. *Additive Manufacturing* 52(February), p. 102691. doi: 10.1016/j.addma.2022.102691.

Qin, J. et al. 2023. Automated Interlayer Wall Height Compensation for Wire Based Directed Energy Deposition Additive Manufacturing. *Sensors* 23(20). doi: 10.3390/s23208498.

Qin, J., Liu, Y. and Grosvenor, R. 2018. Multi-source data analytics for AM energy consumption prediction. *Advanced Engineering Informatics* 38(November), pp. 840–850. doi: 10.1016/j.aei.2018.10.008.

Qin, Y., Qi, Q., Scott, P.J. and Jiang, X. 2019. Status, comparison, and future of the representations of additive manufacturing data. *Computer-Aided Design* 111, pp. 44–64. doi: 10.1016/j.cad.2019.02.004.

Rai, R., Tiwari, M.K., Ivanov, D. and Dolgui, A. 2021. Machine learning in manufacturing and industry 4.0 applications. *International Journal of Production Research* 59(16), pp. 4773–4778. doi: 10.1080/00207543.2021.1956675.

Al Rashid, A., Ahmed, W., Khalid, M.Y. and Koç, M. 2021. Vat photopolymerization of polymers and polymer composites: Processes and applications. *Additive Manufacturing* 47(September), p. 102279. doi: 10.1016/j.addma.2021.102279.

Reed, R. 1993. Pruning algorithms-a survey. *IEEE Transactions on Neural Networks* 4(5), pp. 740–747. doi: 10.1109/72.248452.

Renken, V., von Freyberg, A., Schünemann, K., Pastors, F. and Fischer, A. 2019. In-process closed-loop control for stabilising the melt pool temperature in selective laser melting. *Progress in Additive Manufacturing* 4(4), pp. 411–421. doi: 10.1007/s40964-019-00083-9.

Ribeiro, T.P., Bernardo, L.F.A. and Andrade, J.M.A. 2021. Topology Optimisation in Structural Steel Design for Additive Manufacturing. *Applied Sciences* 11(5), p. 2112. doi: 10.3390/app11052112.

Rodriguez-Andina, J.J., Moure, M.J. and Valdes, M.D. 2007. Features, design tools, and application domains of FPGAs. *IEEE Transactions on Industrial Electronics* 54(4), pp. 1810–1823. doi: 10.1109/TIE.2007.898279.

Rodriguez-Araujo, J., Rodriguez-Andina, J.J., Farina, J., Vidal, F., Mato, J.L. and Montealegre, M.A. 2012. Industrial Laser Cladding Systems: FPGA-Based Adaptive Control. *IEEE Industrial Electronics Magazine* 6(4), pp. 35–46. doi: 10.1109/MIE.2012.2221356.

Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C. and Bengio, Y. 2015. FitNets: Hints for Thin Deep Nets. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–13.

Ronneberger, O., Fischer, P. and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *IEEE Access* 9, pp. 16591–16603.

Saimon, A.I., Yangue, E., Yue, X., Kong, Z.J. and Liu, C. 2024. Advancing Additive Manufacturing through Deep Learning: A Comprehensive Review of Current Progress and Future Challenges. (Dl), pp. 1–45.

Salehinejad, H., Sankar, S., Barfett, J., Colak, E. and Valaee, S. 2017. Recent Advances in Recurrent Neural Networks. pp. 1–21.

Sander, J., Ester, M., Kriegel, H.P. and Xu, X. 1998. Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. *Data Mining and Knowledge Discovery* 2(2), pp. 169–194. doi: https://doi.org/10.1023/A:1009745219419.

Sarker, I.H. 2021. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Computer Science* 2(6), pp. 1–20. doi: 10.1007/s42979-021-00815-1.

Saxena, A. 2022. An Introduction to Convolutional Neural Networks. *International Journal for Research in Applied Science and Engineering Technology* 10(12), pp. 943–947. doi: 10.22214/ijraset.2022.47789.

Scharf, D. et al. 2019. Hardware Accelerated Image Processing on an FPGA-SoC Based Vision System for Closed Loop Monitoring and Additive Manufacturing Process Control. In: *Computer Vision Systems. ICVS 2019*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 3–12. doi: 10.1007/978-3-030-34995-0_1.

Schmitt, J., Bönig, J., Borggräfe, T., Beitinger, G. and Deuse, J. 2020. Predictive model-based quality inspection using Machine Learning and Edge Cloud Computing. *Advanced Engineering Informatics* 45(May 2019), p. 101101. doi: 10.1016/j.aei.2020.101101.

Schohl, A. 2003. A role for maternal -catenin in early mesoderm induction in Xenopus. *The EMBO Journal* 22(13), pp. 3303–3313. Available at: http://emboj.embopress.org/cgi/doi/10.1093/emboj/cdg328.

Seng, K.P., Lee, P.J. and Ang, L.M. 2021. Embedded Intelligence on FPGA: Survey, Applications and Challenges. *Electronics* 10(8), p. 895. doi: 10.3390/electronics10080895.

Seyedzavvar, M. 2023. A hybrid ANN/PSO optimization of material composition and process parameters for enhancement of mechanical characteristics of 3D-printed sample. *Rapid Prototyping Journal* 29(6), pp. 1270–1288. doi: 10.1108/RPJ-10-2022-0338.

Shami, T.M., El-Saleh, A.A., Alswaitti, M., Al-Tashi, Q., Summakieh, M.A. and Mirjalili, S. 2022. Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access* 10, pp. 10031–10061. doi: 10.1109/ACCESS.2022.3142859.

Shao, M., Li, S., Peng, Z. and Sun, Y. 2023. Adversarial-Based Ensemble Feature Knowledge Distillation. *Neural Processing Letters* 55(8), pp. 10315–10329. doi: 10.1007/s11063-023-11328-8.

Sing, S.L. et al. 2017. Direct selective laser sintering and melting of ceramics: a review. *Rapid Prototyping Journal* 23(3), pp. 611–623. doi: 10.1108/RPJ-11-2015-0178.

Singh, R. et al. 2020. Powder bed fusion process in additive manufacturing: An overview. *Materials Today: Proceedings* 26, pp. 3058–3070. doi: 10.1016/j.matpr.2020.02.635.

Singh, R. and Gill, S.S. 2023. Edge AI: A survey. *Internet of Things and Cyber-Physical Systems* 3, pp. 71–92. doi: 10.1016/j.iotcps.2023.02.004.

Sotoodeh, K. 2022. Manufacturing Process. In: *Pipeline Valve Technology*. Boca Raton: CRC Press, pp. 79–97. doi: 10.1201/9781003343318-6.

Srivastava, M., Jayakumar, V., Udayan, Y., M, S., S M, M., Gautam, P. and Nag, A. 2024. Additive manufacturing of Titanium alloy for aerospace applications: Insights into the process, microstructure, and mechanical properties. *Applied Materials Today* 41(July), p. 102481. Available at: https://doi.org/10.1016/j.apmt.2024.102481.

Sturm, L.D., Albakri, M.I., Tarazaga, P.A. and Williams, C.B. 2019. In situ monitoring of material jetting additive manufacturing process via impedance based measurements. *Additive Manufacturing* 28(May), pp. 456–463. doi: 10.1016/j.addma.2019.05.022.

Sun, C., Wang, Y., McMurtrey, M.D., Jerred, N.D., Liou, F. and Li, J. 2021. Additive manufacturing for energy: A review. *Applied Energy* 282(November 2020). doi: 10.1016/j.apenergy.2020.116041.

Sutskever, I., Martens, J. and Hinton, G. 2011. Generating text with recurrent neural networks. In: *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*. pp. 1017–1024.

Tamir, T.S., Xiong, G., Fang, Q., Yang, Y., Shen, Z., Zhou, M. and Jiang, J. 2023. Machine-learning-based monitoring and optimization of processing parameters in 3D printing. *International Journal of Computer Integrated Manufacturing* 36(9), pp. 1362–1378. doi: 10.1080/0951192X.2022.2145019.

Tang, Y. and Zhao, Y.F. 2016. A survey of the design methods for additive manufacturing to improve functional performance. *Rapid Prototyping Journal* 22(3), pp. 569–590. doi: 10.1108/RPJ-01-2015-0011.

Tang, Z. et al. 2020. A review on in situ monitoring technology for directed energy deposition of metals. *The International Journal of Advanced Manufacturing Technology* 108(11–12), pp. 3437–3463. doi: 10.1007/s00170-020-05569-3.

Thoben, K.-D., Wiesner, S. and Wuest, T. 2017. "Industrie 4.0" and Smart Manufacturing – A Review of Research Issues and Application Examples. *International Journal of Automation Technology* 11(1), pp. 4–16. doi: 10.20965/ijat.2017.p0004.

Thompson, M.K. et al. 2016. Design for Additive Manufacturing: Trends, opportunities, considerations, and constraints. *CIRP Annals - Manufacturing Technology* 65(2), pp. 737–760. doi: 10.1016/j.cirp.2016.05.004.

Tian, C., Xu, Y., Zuo, W., Lin, C.-W. and Zhang, D. 2022. Asymmetric CNN for Image Superresolution. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 52(6), pp. 3718–3730. doi: 10.1109/TSMC.2021.3069265.

Tian, W., Ma, J. and Alizadeh, M. 2019a. Energy consumption optimization with geometric accuracy consideration for fused filament fabrication processes. *The International Journal of Advanced Manufacturing Technology* 103(5–8), pp. 3223–3233. doi: 10.1007/s00170-019-03683-5.

Tian, Y., Krishnan, D. and Isola, P. 2019b. Contrastive Representation Distillation. *8th International Conference on Learning Representations, ICLR 2020* (2014), pp. 1–19.

Tiwari, A.S. and Yang, S. 2023. Energy Consumption Modeling of 3D-Printed Carbon-Fiber-Reinforced Polymer Parts. *Polymers* 15(5), p. 1290. doi: 10.3390/polym15051290.

Tran, T.N., Drab, K. and Daszykowski, M. 2013. Revised DBSCAN algorithm to cluster data with dense adjacent clusters. *Chemometrics and Intelligent Laboratory Systems* 120, pp. 92–96. doi: 10.1016/j.chemolab.2012.11.006.

Troussas, C., Krouska, A., Sgouropoulou, C. and Voyiatzis, I. 2020. Ensemble Learning Using Fuzzy Weights to Improve Learning Style Identification for Adapted Instructional Routines. *Entropy* 22(7), p. 735. doi: 10.3390/e22070735.

Ulkir, O. 2023. Energy-Consumption-Based Life Cycle Assessment of Additive-Manufactured Product with Different Types of Materials. *Polymers* 15(6), p. 1466. doi: 10.3390/polym15061466.

Vaneker, T., Bernard, A., Moroni, G., Gibson, I. and Zhang, Y. 2020. Design for additive manufacturing: Framework and methodology. *CIRP Annals* 69(2), pp. 578–599. doi: 10.1016/j.cirp.2020.05.006.

Venieris, S.I. and Bouganis, C.-S. 2017. Latency-driven design for FPGA-based convolutional neural networks. In: *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, pp. 1–8. doi: 10.23919/FPL.2017.8056828.

Vock, S., Klöden, B., Kirchner, A., Weißgärber, T. and Kieback, B. 2019. Powders for powder bed fusion: a review. *Progress in Additive Manufacturing* 4(4), pp. 383–397. doi: 10.1007/s40964-019-00078-6.

Wang, C.-H., Huang, K.-Y., Yao, Y., Chen, J.-C., Shuai, H.-H. and Cheng, W.-H. 2024a. Lightweight Deep Learning: An Overview. *IEEE Consumer Electronics Magazine* 13(4), pp. 51–64. doi: 10.1109/MCE.2022.3181759.

Wang, G., Zhao, P., Shi, Y., Zhao, C. and Yang, S. 2024b. Generative Model-Based Feature Knowledge Distillation for Action Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence* 38(14), pp. 15474–15482. doi: 10.1609/aaai.v38i14.29473.

Wang, K., Yu, L., Xu, J., Zhang, S. and Qin, J. 2022a. Energy consumption intelligent modeling and prediction for additive manufacturing via multisource fusion and inter-layer consistency. *Computers & Industrial Engineering* 173(August), p. 108720. doi: 10.1016/j.cie.2022.108720.

Wang, K., Zhang, Y., Song, Y., Xu, J., Zhang, S. and Tan, J. 2024c. Deep Pattern Matching for Energy Consumption Prediction of Complex Structures in Ecological Additive Manufacturing. *IEEE Transactions on Industrial Informatics* 20(3), pp. 3510–3520. doi: 10.1109/TII.2023.3281649.

Wang, L. and Alexander, C.A. 2016. Additive manufacturing and big data. *International Journal of Mathematical, Engineering and Management Sciences* 1(3), pp. 107–121. doi: 10.33889/ijmems.2016.1.3-012.

Wang, L. and Yoon, K.-J. 2020. Knowledge Distillation and Student-Teacher Learning for Visual Intelligence: A Review and New Outlooks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44(6), pp. 3048–3068. doi: 10.1109/TPAMI.2021.3055564.

Wang, X., Han, Y., Leung, V.C.M., Niyato, D., Yan, X. and Chen, X. 2020a. Convergence of Edge Computing and Deep Learning: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials* 22(2), pp. 869–904. doi: 10.1109/COMST.2020.2970550.

Wang, X., Zhao, Y. and Pourpanah, F. 2020b. Recent advances in deep learning. *International Journal of Machine Learning and Cybernetics* 11(4), pp. 747–750. doi: 10.1007/s13042-020-01096-5.

Wang, Z. et al. 2022b. Data-driven modeling of process, structure and property in additive manufacturing: A review and future directions. *Journal of Manufacturing Processes* 77(February), pp. 13–31. doi: 10.1016/j.jmapro.2022.02.053.

Watson, J.K. and Taminger, K.M.B. 2018. A decision-support model for selecting additive manufacturing versus subtractive manufacturing based on energy consumption. *Journal of Cleaner Production* 176, pp. 1316–1322. doi: 10.1016/j.jclepro.2015.12.009.

Wilt, J.K., Yang, C. and Gu, G.X. 2020. Accelerating Auxetic Metamaterial Design with Deep Learning. *Advanced Engineering Materials* 22(5), pp. 1–7. doi: 10.1002/adem.201901266.

Woo, S., Park, J., Lee, J.-Y. and Kweon, I.S. 2018. CBAM: Convolutional Block Attention Module. In: *Lecture Notes in Computer Science (including subseries Lecture*

Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 3–19. Available at: https://link.springer.com/10.1007/978-3-030-01234-2_1.

Wu, C. et al. 2023a. Topology optimisation for design and additive manufacturing of functionally graded lattice structures using derivative-aware machine learning algorithms. *Additive Manufacturing* 78(May), p. 103833. doi: 10.1016/j.addma.2023.103833.

Wu, D., Jennings, C., Terpenny, J., Gao, R.X. and Kumara, S. 2017. A Comparative Study on Machine Learning Algorithms for Smart Manufacturing: Tool Wear Prediction Using Random Forests. *Journal of Manufacturing Science and Engineering* 139(7), p. 71018. doi: 10.1115/1.4036350.

Wu, H., Judd, P., Zhang, X., Isaev, M. and Micikevicius, P. 2020. Integer Quantization for Deep Learning Inference: Principles and Empirical Evaluation. pp. 1–20.

Wu, R., Liu, B., Fu, P. and Chen, H. 2023b. An efficient lightweight CNN acceleration architecture for edge computing based-on FPGA. *Applied Intelligence* 53(11), pp. 13867–13881. doi: 10.1007/s10489-022-04251-3.

Wu, X. et al. 2008. Top 10 algorithms in data mining. *Knowledge and Information Systems* 14(1), pp. 1–37. doi: 10.1007/s10115-007-0114-2.

Xia, C., Pan, Z., Polden, J., Li, H., Xu, Y. and Chen, S. 2022. Modelling and prediction of surface roughness in wire arc additive manufacturing using machine learning. *Journal of Intelligent Manufacturing* 33(5), pp. 1467–1482. doi: 10.1007/s10845-020-01725-4.

Xie, L., Cen, X., Lu, H., Yin, G. and Yin, M. 2024. A hierarchical feature-logit-based knowledge distillation scheme for internal defect detection of magnetic tiles. *Advanced Engineering Informatics* 61(April), p. 102526. doi: 10.1016/j.aei.2024.102526.

Xie, Q., Luong, M.-T., Hovy, E. and Le, Q. V. 2020. Self-Training With Noisy Student Improves ImageNet Classification. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 10684–10695. doi: 10.1109/CVPR42600.2020.01070.

Xin, X., Song, H. and Gou, J. 2024. A New Similarity-Based Relational Knowledge Distillation Method. In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 3535–3539. doi: 10.1109/ICASSP48485.2024.10447596.

Xu, C., Jiang, S., Luo, G., Sun, G., An, N., Huang, G. and Liu, X. 2022. The Case for FPGA-Based Edge Computing. *IEEE Transactions on Mobile Computing* 21(7), pp. 2610–2619. doi: 10.1109/TMC.2020.3041781.

Xu, J., Wang, K., Sheng, H., Gao, M., Zhang, S. and Tan, J. 2020. Energy efficiency optimization for ecological 3D printing based on adaptive multi-layer customization. *Journal of Cleaner Production* 245, p. 118826. doi: 10.1016/j.jclepro.2019.118826.

Yan, Z. et al. 2022. A New Method of Predicting the Energy Consumption of Additive Manufacturing considering the Component Working State. *Sustainability* 14(7), p. 3757. doi: 10.3390/su14073757.

Yang, C., Xie, L., Qiao, S. and Yuille, A.L. 2019. Training Deep Neural Networks in Generations: A More Tolerant Teacher Educates Better Students. *Proceedings of the AAAI Conference on Artificial Intelligence* 33(01), pp. 5628–5635. doi: 10.1609/aaai.v33i01.33015628.

Yang, G., Tang, Y., Wu, Z., Li, J., Xu, J. and Wan, X. 2024a. DMKD: Improving Feature-Based Knowledge Distillation for Object Detection Via Dual Masking Augmentation. In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 3330–3334. doi: 10.1109/ICASSP48485.2024.10446978.

Yang, G., Yu, S., Sheng, Y. and Yang, H. 2023. Attention and feature transfer based knowledge distillation. *Scientific Reports* 13(1), p. 18369. doi: 10.1038/s41598-023-43986-y.

Yang, Y., He, M. and Li, L. 2020a. Power consumption estimation for mask image projection stereolithography additive manufacturing using machine learning based approach. *Journal of Cleaner Production* 251, p. 119710. doi: 10.1016/j.jclepro.2019.119710.

Yang, Y., Li, L., Pan, Y. and Sun, Z. 2017. Energy Consumption Modeling of Stereolithography-Based Additive Manufacturing Toward Environmental Sustainability. *Journal of Industrial Ecology* 21.

Yang, Y., Yang, R., Pan, L., Ma, J., Zhu, Y., Diao, T. and Zhang, L. 2020b. A lightweight deep learning algorithm for inspection of laser welding defects on safety vent of power battery. *Computers in Industry* 123, p. 103306. doi: 10.1016/j.compind.2020.103306.

Yang, Z., Li, Z., Zeng, A., Li, Z., Yuan, C. and Li, Y. 2024b. ViTKD: Feature-based Knowledge Distillation for Vision Transformers. In: *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, pp. 1379–1388. doi: 10.1109/CVPRW63382.2024.00145.

Yao, J., Luo, X., Li, F., Li, J., Dou, J. and Luo, H. 2024. Research on hybrid strategy Particle Swarm Optimization algorithm and its applications. *Scientific Reports* 14(1), p. 24928. doi: 10.1038/s41598-024-76010-y.

Yap, Y.L., Wang, C., Sing, S.L., Dikshit, V., Yeong, W.Y. and Wei, J. 2017. Material jetting additive manufacturing: An experimental study using designed metrological benchmarks. *Precision Engineering* 50, pp. 275–285. doi: 10.1016/j.precisioneng.2017.05.015.

Yehia, H.M., Hamada, A., Sebaey, T.A. and Abd-Elaziem, W. 2024. Selective Laser Sintering of Polymers: Process Parameters, Machine Learning Approaches, and Future Directions. *Journal of Manufacturing and Materials Processing* 8(5). Available at: https://www.mdpi.com/2504-4494/8/5/197.

Yi, L., Ravani, B. and Aurich, J.C. 2019. Development of a simulation tool for predicting energy consumption of selective laser melting by using MATLAB/Simulink. *Procedia CIRP* 81(March), pp. 28–33. doi: 10.1016/j.procir.2019.03.006.

Yim, J., Joo, D., Bae, J. and Kim, J. 2017. A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 7130–7138. doi: 10.1109/CVPR.2017.754.

El youbi El idrissi, M.A., Laaouina, L., Jeghal, A., Tairi, H. and Zaki, M. 2023. Modeling of Energy Consumption and Print Time for FDM 3D Printing Using Multilayer Perceptron Network. *Journal of Manufacturing and Materials Processing* 7(4), p. 128. doi: 10.3390/jmmp7040128.

Yuan, Z., Yang, Z., Ning, H. and Tang, X. 2024. Multiscale knowledge distillation with attention based fusion for robust human activity recognition. *Scientific Reports* 14(1), pp. 1–16. doi: 10.1038/s41598-024-63195-5.

Zagoruyko, S. and Komodakis, N. 2016. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pp. 1–13.

Zhang, C., Li, P., Sun, G., Guan, Y., Xiao, B. and Cong, J. 2015. Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks. In: *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. FPGA '15. New York, NY, USA, NY, USA: ACM, pp. 161–170. doi: 10.1145/2684746.2689060.

Zhang, F. et al. 2021. The recent development of vat photopolymerization: A review. *Additive Manufacturing* 48(October), p. 102423. doi: 10.1016/j.addma.2021.102423.

Zhang, J., Li, D. and Wang, Y. 2020. Toward intelligent construction: Prediction of mechanical properties of manufactured-sand concrete using tree-based models. *Journal of Cleaner Production* 258, p. 120665. doi: https://doi.org/10.1016/j.jclepro.2020.120665.

Zhang, S., Liu, H. and He, K. 2024. Knowledge Distillation via Token-Level Relationship Graph Based on the Big Data Technologies. *Big Data Research* 36(January), p. 100438. doi: 10.1016/j.bdr.2024.100438.

Zhang, Y., Hong, G.S., Ye, D., Zhu, K. and Fuh, J.Y.H. 2018a. Extraction and evaluation of melt pool, plume and spatter information for powder-bed fusion AM process monitoring. *Materials & Design* 156, pp. 458–469. doi: 10.1016/j.matdes.2018.07.002.

Zhang, Y., Safdar, M., Xie, J., Li, J., Sage, M. and Zhao, Y.F. 2023a. A systematic review on data of additive manufacturing for machine learning applications: the data quality, type, preprocessing, and management. *Journal of Intelligent Manufacturing* 34(8), pp. 3305–3340. doi: 10.1007/s10845-022-02017-9.

Zhang, Y., Xiang, T., Hospedales, T.M. and Lu, H. 2018b. Deep Mutual Learning. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 4320–4328. doi: 10.1109/CVPR.2018.00454.

Zhang, Z.-J. et al. 2023b. Recommendation of SLM Process Parameters Based on Analytic Hierarchy Process and Weighted Particle Swarm Optimization for High-Temperature Alloys. *Materials* 16(16), p. 5656. doi: 10.3390/ma16165656.

Zhu, B., Zhang, W., Zhang, W. and Li, H. 2023. Generative design of texture for sliding surface based on machine learning. *Tribology International* 179(November 2022), p. 108139. doi: 10.1016/j.triboint.2022.108139.

Zhu, Z. and Zhang, W. 2024. Exploring Feature-based Knowledge Distillation For Recommender System: A Frequency Perspective. 1(1).

Zuccarini, C., Ramachandran, K. and Jayaseelan, D.D. 2024. Material discovery and modeling acceleration via machine learning. *APL Materials* 12(9). Available at: https://doi.org/10.1063/5.0230677.

Appendix A Preliminary Study on Energy Consumption Prediction Modelling

A.1 Machine Learning

- ***Support Vector Regression (SVR)***

Support Vector Machine (SVM) includes one or multiple hyperplanes in a high- or infinite-dimensional space (Wu et al. 2017). It can realise classification and regression. SVM refers to finding the best classification function to divide samples into two classes in training data in a two-class learning task. Because of the constraints of the training data or noise, the samples outside the training data might be closer to the boundaries, which makes the hyperplane present an incorrect division. The maximum margin hyperplane has the least impact, i.e. the classification result of that is the most robust (Wu et al. 2008). Equation A.1 expresses the division of the hyperplane described by following a linear equation to determine the position of the hyperplane.

$$f(x) = \boldsymbol{\omega}^T \boldsymbol{x} + b \tag{A.1}$$

where vector $\boldsymbol{\omega}$ and constant $\boldsymbol{b}$ construct the linear equation of division of hyperplane.

Support Vector Regression (SVR) will use different kernel methods that satisfy Mercer's theorem to solve the convex problem. Some prevalent kernel involves linear, polynomial, Gaussian Radial Basis Function (RBF) and sigmoid (Wu et al. 2017). The model can be represented in the following form:

$$\omega = \sum_{i=1}^{m} (\hat{\alpha}_i + \alpha_i)\phi(x_i) \quad (A.2)$$

$$f(x) = \sum_{i=1}^{m} (\hat{\alpha}_i + \alpha_i)\kappa(x, x_i) + b \quad (A.3)$$

where $\boldsymbol{\kappa(x_i, x_j)} = \boldsymbol{\phi(x_i)^T \phi(x_j)}$ expresses the kernel function.

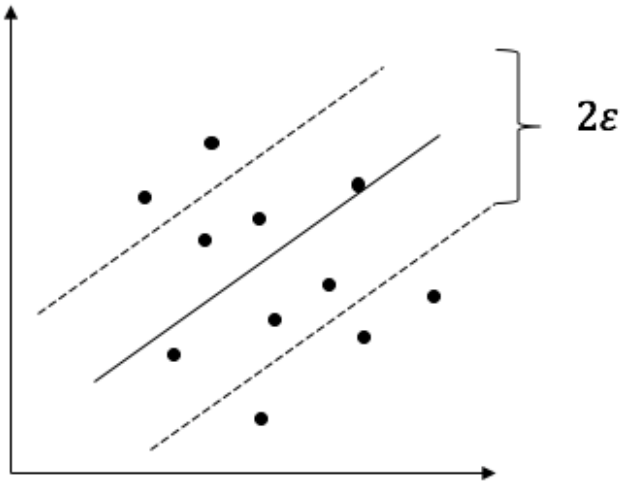


Figure A. 1 Divided hyperplanes to separate the two types of training samples.

As shown in Figure A. 1, $\boldsymbol{f(x)}$ (the solid line) can be regarded as the centre and used to construct an interval band with a bandwidth of $\boldsymbol{2\varepsilon}$. If the training samples fall into it, the prediction is considered correct.

- ***Gradient Boost Regression Tree (GBRT)***

Gradient Boosting Regression Tree (GBRT) or known as Gradient Boosting Decision Tree (GBDT) is a way of ensemble learning, which integrates weak learners sequentially, each trying to correct respective predecessors, to form a strong learner. In general, DT is applied in this method, and this tree-based ensemble method will provide improved performance. It employs an iterative tree-based algorithm that consists of various decision trees, and the result of these trees is combined to obtain the final result. As same as SVM, it has a strong generalisation ability. It uses gradient boosting to avoid the overfitting problem. GBRT shows outstanding performance in processing various features, predictive modelling, and processing outliers by the loss function (Zhang et al. 2020).

For each weak learner $\boldsymbol{h}$, the model $\boldsymbol{F_m}$ consists of regression trees, and the previous preceding model from each iteration accumulates the new regression tree model during the entire learning process.

$$F_{m+1}(x) = F_m(x) + h(x) \tag{A. 4}$$

When introducing a training set $\{(\boldsymbol{x_1}, \boldsymbol{y_1}), (\boldsymbol{x_2}, \boldsymbol{y_2}), \dots, (\boldsymbol{x_n}, \boldsymbol{y_n})\}$ to the model, GBRT will improve the performance by minimising its loss function $\boldsymbol{L(y_i, \gamma)}$ with initialised model:

$$F_0(x) = argmin \sum_{i=1}^{n} L(y_i, \gamma) \tag{A. 5}$$

The updated model will be:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{A. 6}$$

where $\boldsymbol{\gamma_m = argmin \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))}$ for $\boldsymbol{m}$ regression trees.

- ***Density-Based Spatial Clustering of Applications with Noise (DBSCAN)***

The clustering problem is related to an unsupervised learning problem. According to predefined rules, the clustering problem is used to find the uncovered patterns to be classified with similar characteristics between data (Jiang et al. 2011). Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a data clustering algorithm targeting unstructured data (Ester et al. 1996). Specifically, DBSCAN used a density-based clustering approach, which is the most used in clustering spatial data. This algorithm adopts the concept of density-based clustering, which requires the number of points in a specific region of the clustering space, with minimum numbers of objects $\boldsymbol{MinPts}$ and should exceed the given threshold. The following equations demonstrates the nature of DBSCAN, and the random point p in its neighbourhood is defined in equation (A.7)

$$N_{Eps} = \left\{ q \in \frac{D}{dist}(p, q) < Eps \right\} \quad (A.7)$$

$$N_{Eps}(P) > MinPts \quad (A.8)$$

where Eps is the neighbourhood of the radius, given the collection of objects D. The core point $\boldsymbol{P}$ is defined in equation (A.8) if it contains a minimal number of points. In other words, a core point, a boundary point or an outlier is determined by two indicators: $\boldsymbol{MinPts}$ and $\boldsymbol{Eps}$, and the outlier is removed. The algorithm connects core points under equation (A.8), allocating the boundary point to the closest core point and finally obtaining the clustering results (Jiang et al. 2011).

Compared to k-means clustering, DBSCAN shows faster clustering speed and effectiveness in processing noise points, handling abnormal data, and exploring spatial clusters of random shapes. Besides, the unbiased-shaped clusters do not need to divide

the number of clusters (Sander et al. 1998; Birant and Kut 2007). A satisfactory clustering algorithm needs to have the following characteristics: 1) to determine knowledge from inputs, especially for large datasets, 2) capable of finding arbitrarily shaped clusters and 3) efficient in handling large datasets (Tran et al. 2013). The working environment data is collected layer by layer over thousands of data in separate files with various types from the entire process because of large data volume and heterogeneity. Therefore, DBSCAN is expected to tackle the issues. Furthermore, this algorithm was applied at the beginning, demonstrating the mean values, which can be representative of the entire cluster. These values can be combined into design-relevant datasets on the build level to unify the format of each working environment data file.

- ***Extreme Gradient Boost (XGBoost)***

Extreme Gradient Boost (XGBoost) refers to a tree-based ensemble learning using a tree algorithm proposed by Chen and Guestrin (Chen and Guestrin 2016). This boosting method is an effective ML method. XGBoost uses regression tree ensembles with the same decision rules as the decision tree (DT) and one score for each leaf value. Two aspects allow it to be distinguished from other tree-boosting machines. Firstly, XGBoost has a different objective function. For each regression tree, this ensemble method accumulates the sum of scores as the prediction value for all trees. Assuming there are k trees, the output for the tree ensemble is defined as follows:

$$\hat{y}_i = \sum_{k=1}^{K} f_x(x_i), f_x \in \mathcal{F} \quad \text{(A. 9)}$$

The objective function is the sum of training loss and complexity of the trees to control overfitting, and it is:

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k}^{K} \Omega(f_k), f_x \in \mathcal{F} \quad \text{(A. 10)}$$

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|w\|^2 \quad \text{(A. 11)}$$

where $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$ is the predicted value of the model, y_i stands for the ith feature label, $\boldsymbol{f}_{\boldsymbol{k}}$ represents the k-th tree model, $\boldsymbol{T}$ is the number of nodes and $\boldsymbol{w}$ is the collection of score combinations. In reducing the objective function, the predicted value adds a new function $\boldsymbol{f}$ in each iteration. This additive training defines a new objective function to optimise and search for a new tree model.

Another difference is the division of nodes. There are four proposed splitting algorithms from Chen and Guestrin's work. XGBoost adopts (1) a basic exact greedy algorithm, (2) an approximate algorithm, (3) a weighted quantile sketch and (4) sparsity-aware split-finding methods. Among these four split-finding algorithms, algorithms (2) and (3) solve the problem of the data failing to load into memory at once or algorithm (1) not being distributed efficiently. The XGBoost approach calculates the gain of each feature in parallel and chooses the feature with the most significant information gain to split.

XGBoost provides an idea for processing sparse data and enables handling instance weights in tree learning. Compared with the traditional tree model, it shows the merits of regularisation in controlling the model complexity and reducing the variance of the model to avoid overfitting. This model is used to predict the energy consumption in the SLS system. By targeting this specific task, XGBoost integrates the weak learner to form a stronger learner to increase accuracy. In addition, the sparsity-aware split-finding method of XGBoost can process the missing values in the combined datasets. Also, it increases the learning rate effectively by controlling the model complexity, which is essential when dealing with large datasets.

A.2 A Hybrid Machine Learning Approach for Energy Consumption Prediction on Layer-level Data

The original data was collected from the target AM system where the data can be categorised into four different types. They are operation process, material, working environment and design. In Figure A. 2, process data stem from the parameter settings collected from the SLS machine, such as the measured values from the dispenser, recoater speed and the laser power used in sintering, which relies on the experience and knowledge of the operators. With regards to material data, it depends on the material itself. In this case, the type of material is known, referring to two kinds of nylon powder. Design data is the data collected from computer-aided design (CAD) models created by designers, often including design parameters for each layer (Yang et al. 2017), which are often determined at the beginning stage of the entire process. The working environment can be monitored by sensors and data stored in the conditional monitoring files for the illustration. This kind of data source collected from the working environment by an IoT platform is considered as the layer level from real-time monitoring. Some monitoring files can demonstrate these data to better comprehend the structure of the data.

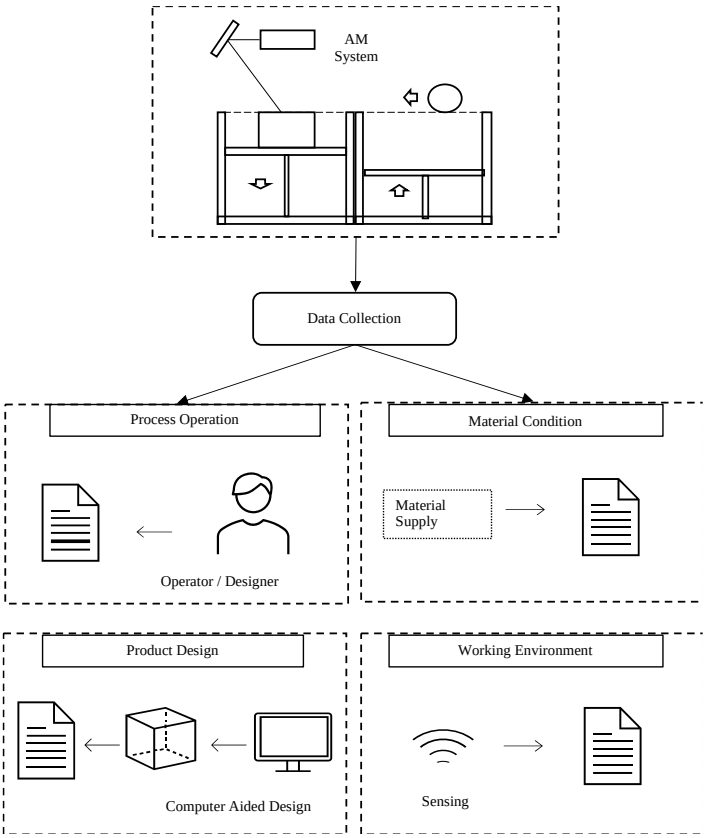


Figure A. 2 The Multi-Source Data Collection from AM System.

Figure A. 3 demonstrates the framework of the proposed methodology in pre-processing and predictive modelling. The entire process can be divided into three stages, corresponding to their respective roles. In the first stage, the input data were collected from the SLS system and categorised into four datasets according to their sources. The working environment data contains different quantities in each separate file, which is essential for integrating these data using DBSCAN to unify the structure of layer-level data. Secondly, the integrated datasets reduced the dimensionality through DBSCAN clustering and combining into the XGBoost decision tree. Finally, energy consumption was obtained, and RMSE and MCC were used to evaluate the performance of XGBoost.

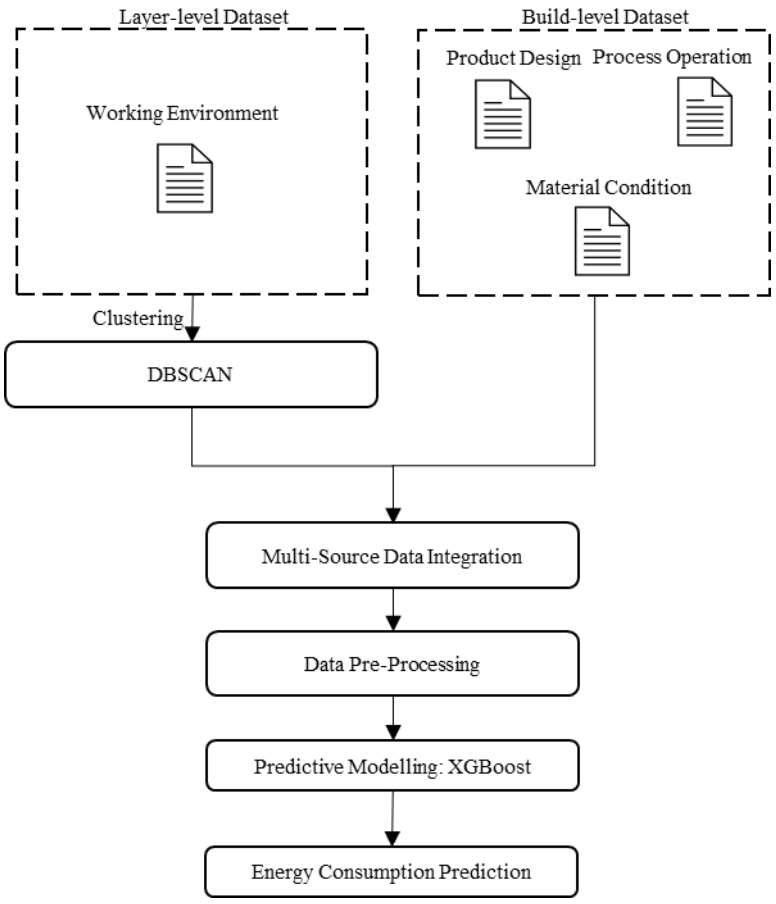


Figure A. 3 The Framework of Proposed Methodology for Energy Consumption Prediction.

Advanced data analysis and ML methods show the ability to predict energy consumption. This work utilised a hybrid approach that combines unsupervised and supervised learning, where unsupervised learning aimed to integrate different dimensional datasets, while supervised learning was utilised to predict energy consumed in the SLS system.

In the case study, three ML algorithms and one DL technique were implemented to predict energy consumption based on working environment data combined with various other datasets. The employed ML algorithms include XGBoost, SVR, and GBRT, alongside CNN.

The effectiveness of these methods was evaluated using metrics such as the RMSE and MCC. XGBoost demonstrated superior performance, achieving the highest MCC of 0.708 when using combined datasets, indicating a strong positive correlation and fit to the experimental data, particularly after applying DBSCAN clustering. The MCC values for SVR and GBRT were comparably high (0.669 and 0.676, respectively), suggesting that these models are also effective for predictive tasks. Conversely, CNN, typically used for image data and classification tasks, is less common in regression settings and performed less optimally in this study. When multi-source data were utilised, ensemble methods like GBRT and XGBoost optimized performance, whereas other methods experienced slight decreases in MCC. The RMSE values provided further insight into model accuracy, with XGBoost showing the lowest error at 130.783 Wh/g, indicating the minor deviation between actual and predicted values across all datasets. CNN displayed a higher RMSE of 231.958 Wh/g, reflecting its less typical use in regression tasks within industry settings where pattern recognition or classification is common.

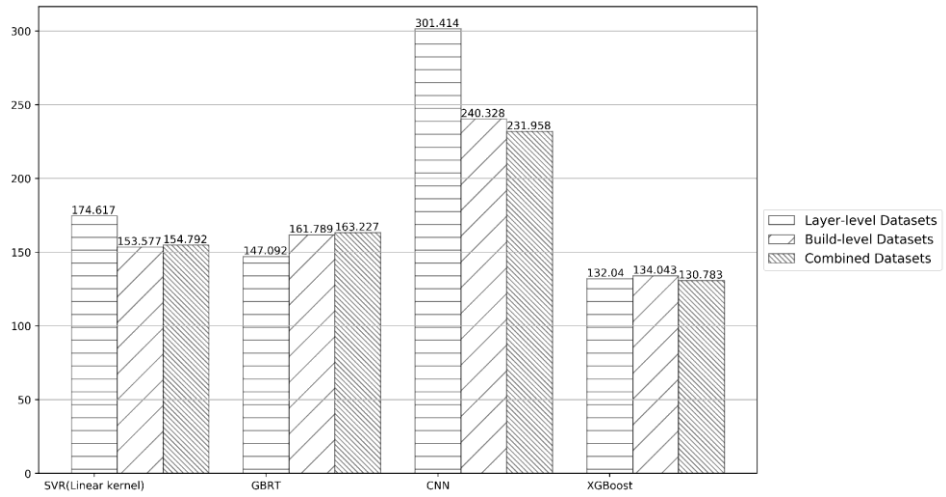


Figure A. 4 Comparison of RMSE of XGBoost and benchmarks.

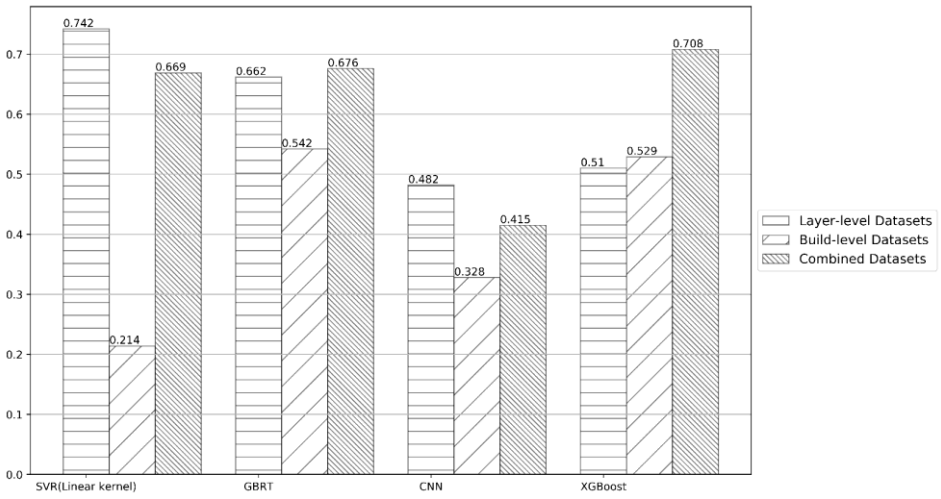


Figure A. 5 Comparison of MCC of XGBoost and benchmarks.

Figure A. 4 and Figure A. 5 from the study illustrate the RMSE and MCC comparisons, and Figure A. 6 displays the alignment between predicted and actual data, indicating a general trend in energy consumption prediction despite outliers. Integrating heterogeneous data into the XGBoost model revealed fluctuating patterns, suggesting a gap between predicted and actual data that may be attributed to irrelevant features in the datasets. This issue could be addressed by refining data collection processes to generate new, more relevant features.

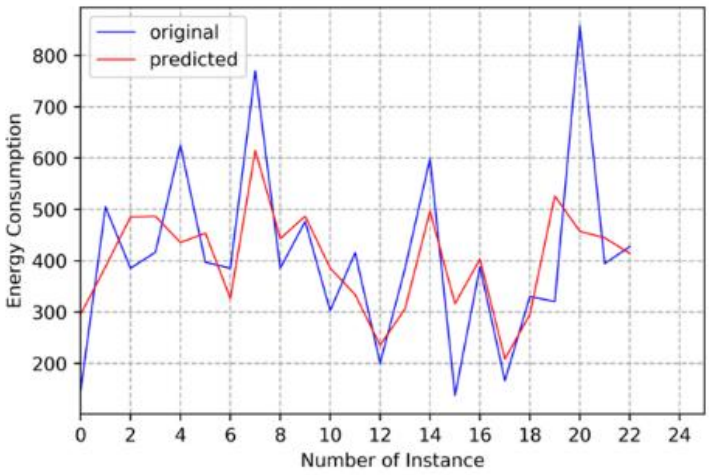


Figure A. 6 The prediction result between predicted values and original values.

This hybrid ML approach has presented better performance than the other three algorithms and connected the target and input with high dimensionality when using combined datasets. A single learner cannot be adopted to tackle the real-world issue of handling heterogeneous data while integrating DBSCAN and XGBoost suits the case.

A.3 Particle Swarm Optimisation in Additive Manufacturing

PSO algorithms play a crucial role in AM technologies, which are used to improve manufacturing quality, reduce production time and lower costs. What we know about PSO in AM is largely based upon experimental and case studies that investigate how optimising printing parameters improves the mechanical properties of the part and the efficiency of the manufacturing process. The section below reviews the methodology of different researchers in PSO.

It is believed that the key challenge in AM is to determine the optimal parameters to improve the quality of the final product. PSO is a population-based optimisation algorithm and has been widely used in this case due to its simplicity and fewer control parameters. The literature on PSO algorithms has highlighted several variants to address premature convergence issues. Those methods were widely applied in different optimisation scenarios (Shami et al. 2022). By drawing on the concept of PSO, Yao et al. (2024) have been able to propose a Hybrid Strategy PSO (HSPSO) by integrating multiple strategies. The results demonstrated the superiority and effectiveness of the HSPSO algorithm (Yao et al. 2024). Pathak and Singh provided an in-depth analysis of the work of the PSO-based approach for minimising geometric dimension and tolerance to improve the geometric accuracy of the objects (Pathak and Singh 2017). In their introduction to a new optimisation approach, Zhang et al. (2023) proposed a method combining the Analytical Hierarchy Process (AHP) and Weighted Particle Swarm Optimisation (WPSO) for advising SLM process parameters of high-temperature alloys (Zhang et al. 2023b).

Although the original PSO algorithm showed its merits of optimisation performance, it still suffers from the problem of premature convergence. To address this problem, a case-study approach was chosen to allow a deeper insight into modelling and predicting surface roughness in Wire Arc Additive Manufacturing (WAAM). This method is particularly useful in studying adaptive neuro-fuzzy inference systems to improve the prediction of surface roughness (Xia et al. 2022). Lastly, the work of Xia et al. (2022) on leveraging ML-based approaches to model and predict surface

roughness in WAAM further highlighted the versatility of PSO algorithms, as did the recent study by Murat et al. (2023), which investigated the use of PSO-based response surface methodology to determine optimal SLM process parameters (Murat et al. 2023). Collectively, these studies outline a critical role for PSO and its variants in optimising parameters and mechanical properties.

Up to now, several studies have investigated the effects of PSO algorithms in AM systems in terms of vibration performance optimisation and automatic parameter optimisation strategies. The study by Ekerer et al. (2024) makes an important contribution to the vibration performance of 3D printed cantilever beams by integrating hybrid Artificial Neural Networks (ANNs) and PSO methods to develop a high-precision predictive model (Ekerer et al. 2024). To further investigate the role of this combination, Seyedzavvar carried out a study that leveraged ANN and PSO to optimise process parameters, and the mechanical properties of 3D printed parts with $CaCO_3$ nano additives (Seyedzavvar 2023). Fang and associates demonstrated the potential of a unique PSO algorithm in practical industrial data analytics by introducing it for anomaly identification in Wire Arc Additive Manufacturing (WAAM) (Fang et al. 2024). Sugianto and Kim presented a mixed-integer linear programming model and heuristic PSO-based methods to handle a crucial integrated scheduling problem involving batch AM operations and direct shipping deliveries (Chandra Sugianto and Soo Kim 2024). Liu et al. (2024) presented an automated optimisation strategy for Laser Powder Bed Fusion (LPBF) parameters, utilising a Neural Network (NN)-based infrared radiation intensity prediction model. This strategy effectively mitigated abnormal infrared radiation intensity values during scanning, thus enhancing the reliability of LPBF performance for complex components (Liu et al. 2024).

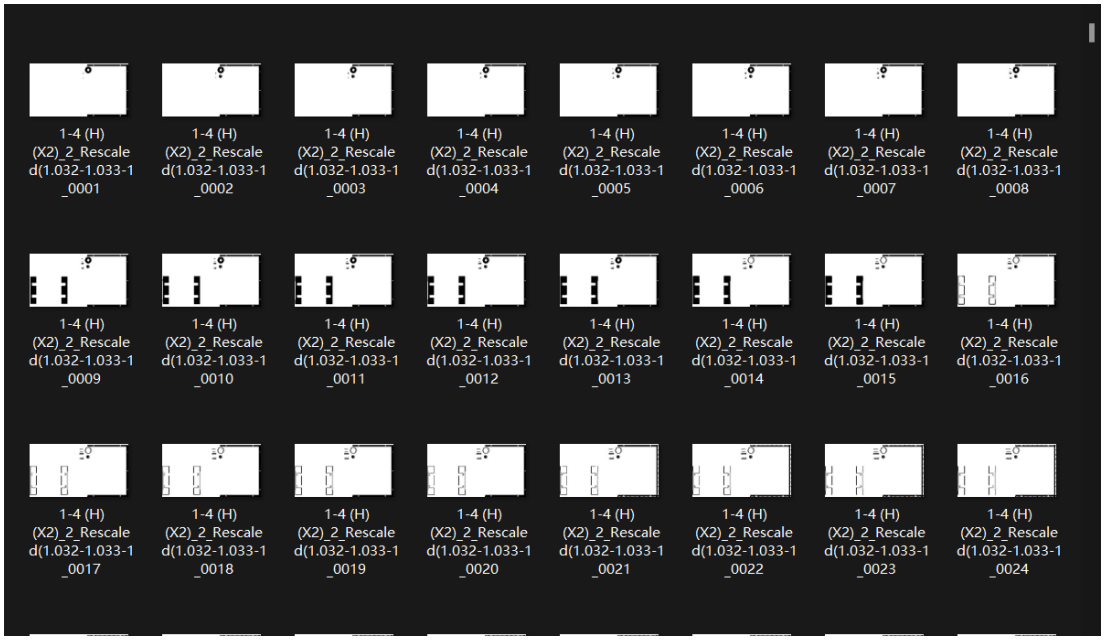
Appendix B Datasets Used in This Thesis

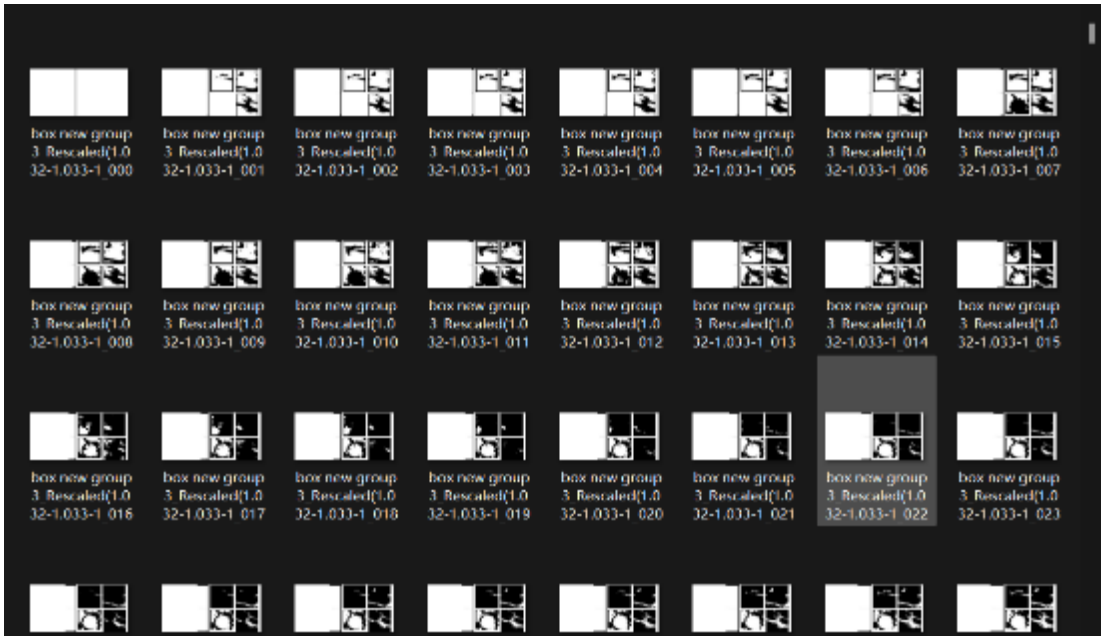
B.1 Screenshot of Layer-wise Image Data for Training (Partially)


566_001_1000_06 - Issue 1_1_Rescaled(1.032-1.033-1_...

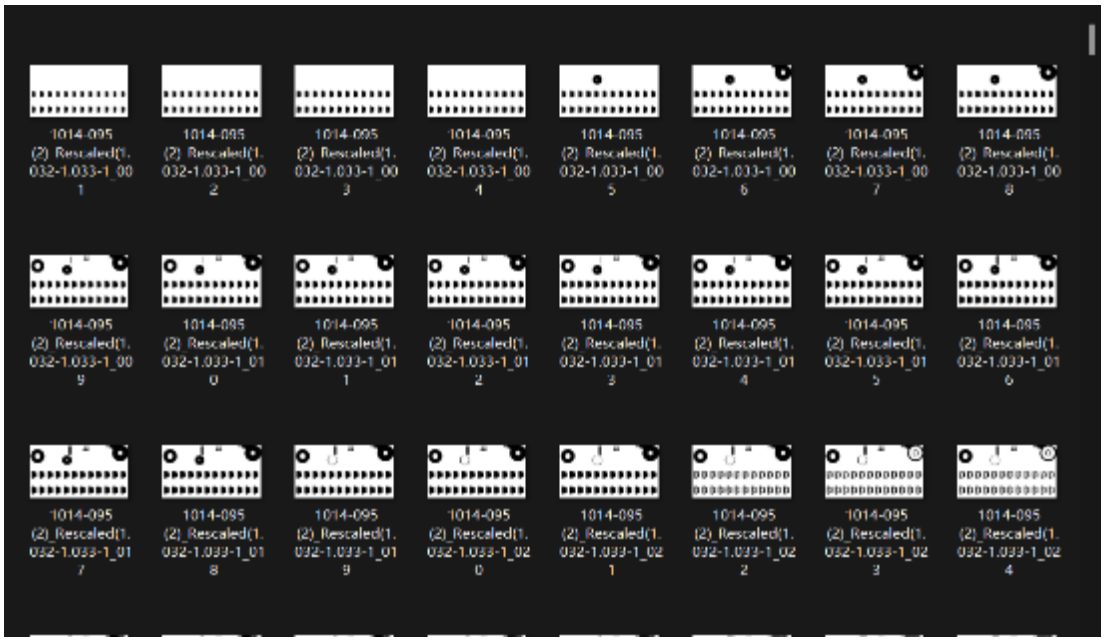

566_001_1000_06 - Issue 1_Rescaled(1.032-1.033-1_0...

B.2 Screenshot of Layer-wise Image Data for Testing (Partially)






All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_001
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_002
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_003
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_004
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_005
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_006
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_007
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_008
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_009
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_010
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_011
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_012
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_013
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_014
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_015
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_016
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_017
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_018
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_019
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_020
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_021
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_022
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_023
All parts.inp_1_Fib_ Rescaled(1.033-1.032-1_024


1014-095 (2) Rescaled(1.032-1.033-1_001
1014-095 (2) Rescaled(1.032-1.033-1_002
1014-095 (2) Rescaled(1.032-1.033-1_003
1014-095 (2) Rescaled(1.032-1.033-1_004
1014-095 (2) Rescaled(1.032-1.033-1_005
1014-095 (2) Rescaled(1.032-1.033-1_006
1014-095 (2) Rescaled(1.032-1.033-1_007
1014-095 (2) Rescaled(1.032-1.033-1_008
1014-095 (2) Rescaled(1.032-1.033-1_009
1014-095 (2) Rescaled(1.032-1.033-1_010
1014-095 (2) Rescaled(1.032-1.033-1_011
1014-095 (2) Rescaled(1.032-1.033-1_012
1014-095 (2) Rescaled(1.032-1.033-1_013
1014-095 (2) Rescaled(1.032-1.033-1_014
1014-095 (2) Rescaled(1.032-1.033-1_015
1014-095 (2) Rescaled(1.032-1.033-1_016
1014-095 (2) Rescaled(1.032-1.033-1_017
1014-095 (2) Rescaled(1.032-1.033-1_018
1014-095 (2) Rescaled(1.032-1.033-1_019
1014-095 (2) Rescaled(1.032-1.033-1_020
1014-095 (2) Rescaled(1.032-1.033-1_021
1014-095 (2) Rescaled(1.032-1.033-1_022
1014-095 (2) Rescaled(1.032-1.033-1_023
1014-095 (2) Rescaled(1.032-1.033-1_024

B.3 The Unit Energy Consumption of Each Layer (Partially)

	A
1	Label
2	3.909894
3	3.851627
4	3.702557
5	3.644118
6	3.374862
7	3.458002
8	3.598884
9	3.566481
10	3.642834
11	3.650497
12	3.522178
13	3.686313
14	3.751038
15	3.409005
16	3.648347
17	3.60854
18	3.953557
19	3.670206
20	3.651923
21	3.724967
22	13.36952
23	15.02358
24	16.77895
25	15.81228
26	17.37456
27	17.12126
28	19.48189
29	17.7486

	A
1	Label
2	17.08781
3	12.59608
4	10.69427
5	9.928762
6	9.144042
7	4.389427
8	4.466981
9	4.068754
10	4.099995
11	4.250906
12	4.174183
13	4.316173
14	4.286035
15	4.266642
16	4.303474
17	4.563557
18	4.532435
19	4.520122
20	4.574571
21	4.728052
22	4.536964
23	4.356234
24	4.445489
25	4.526632
26	4.389599
27	4.637362
28	4.702816
29	4.710085

	A
1	Label
2	3.191119
3	3.203338
4	3.128449
5	3.030026
6	3.131446
7	3.173629
8	2.889246
9	3.145294
10	3.104012
11	3.113576
12	3.214795
13	3.238325
14	3.232468
15	3.313936
16	3.292988
17	3.223005
18	3.413798
19	3.192417
20	3.219612
21	3.408915
22	3.360508
23	3.425921
24	3.586811
25	3.490821
26	3.747641
27	3.599707
28	3.747139
29	3.662508

	A
1	Label
2	16.06894
3	12.42312
4	10.60636
5	8.606837
6	8.658173
7	8.580758
8	7.93038
9	7.457205
10	6.659972
11	6.222406
12	6.489115
13	6.545561
14	6.589169
15	5.93968
16	6.440378
17	6.237364
18	6.03158
19	5.698592
20	5.759005
21	6.419093
22	6.486145
23	6.50058
24	6.791704
25	6.834793
26	6.768273
27	6.826824
28	6.330325
29	7.169099

B.4 The Design-relevant Parameters of the Build

Part filling	PartRate_V	PartRate_H	PartRate_V	Part heigh	Total fillin	TotalRate_	TotalRate_	TotalRate_	Bottom_ar	Heigh	NumPart	Energy
37.60655	0.321654	0.503971	0.63824	52.55192	8.367801	0.52798	0.296037	1.783491	2512.167	204.2029	81	411.1616
36.45841	1.312324	0.098682	13.29849	15.4958	15.65833	0.481711	0.193077	2.494919	2110.806	127.8089	38	151.5222
28.50364	1.161269	0.318051	3.65121	27.24959	15.15736	0.517404	0.152826	3.385573	2316.88	102.2667	46	137.4275